

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN  
METEOROLÓGICA REMOTA DE BAJO COSTO EMPLEANDO EL ENFOQUE DE  
INTERNET DE LAS COSAS

Daniel Steven Murcia Almanza

Anthony Rojas Paternina

UNIVERSIDAD PILOTO DE COLOMBIA  
FACULTAD DE INGENIERÍA  
PROGRAMA INGENIERA DE TELECOMUNICACIONES

2015

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE ESTACIÓN  
METEOROLÓGICA REMOTA DE BAJO COSTO EMPLEANDO EL ENFOQUE DE  
INTERNET DE LAS COSAS.

Daniel Steven Murcia Almanza

Anthony Rojas Paternina

SEMILLERO DE INVESTIGACION SMARTIC

Director

Henry Arturo Bastidas Mora

UNIVERSIDAD PILOTO DE COLOMBIA

FACULTAD DE INGENIERÍA

PROGRAMA INGENIERA DE TELECOMUNICACIONES

2015

Nota de aceptación:

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Bogotá D.C., 30 de Septiembre del 2015.

## **DEDICATORIA**

Este trabajo lo dedicamos a nuestros padres que siempre han estado apoyándonos incondicionalmente en cada etapa de nuestras vidas. También es a los profesores, compañeros y amigos, que han formado parte de nuestro proceso de formación personal y profesional.

## **AGRADECIMIENTOS**

Presentamos nuestros más sinceros agradecimientos a:

Nuestro tutor el ingeniero Henry Arturo Bastidas Mora, quien con sus conocimientos, motivación y experiencia en el área, dedico parte de su tiempo para guiarnos en el proceso de investigación necesario para lograr este proyecto.

Nuestra compañera Leidy Viviana Carvajal Hurtado, que siempre estuvo dispuesta, a brindarnos su ayuda y colaboración; nos apoyó incondicionalmente con sus conocimientos, fortaleciendo para el desarrollo del proyecto.

Nelson Gabriel Liberato Robayo, consejero fundamental para el desarrollo de este proyecto.

Nuestros padres Elisa Mercedes Paternina Vásquez, Luis Javier Murcia García y Lucila Almanza Opayome, por ser la base fundamental para la realizacion de nuestras metas y sueños.

## CONTENIDO

INTRODUCCIÓN .....	11
1. MARCO REFERENCIAL .....	12
1.1 ANTECEDENTES.....	12
1.1.1 Estaciones meteorológicas que usan hardware libre en el mundo. ....	12
1.1.2 Estaciones meteorológicas utilizando hardware libre en Colombia.....	15
1.1.3 El cambio climático en Colombia.....	17
1.1.4 Transmisión de información. ....	19
2. IDENTIFICACIÓN DE VARIABLES FÍSICAS A MEDIR .....	21
2.1 IDENTIFICACIÓN DE VARIABLES FÍSICAS A MEDIR.....	21
2.1.1 Temperatura. ....	21
2.1.2 Humedad. ....	22
2.1.3 Presión atmosférica.....	23
2.1.4 Velocidad del viento. ....	24
2.1.5. La lluvia.....	25
3. DISEÑO DEL SISTEMA DE ADQUISICIÓN, PROCESAMIENTO Y TRANSMISIÓN DE LAS VARIABLES FÍSICAS SENSADAS .....	27
3.1 SENSOR DE HUMEDAD Y TEMPERATURA.....	27
3.2 SENSOR DE PRESIÓN ATMOSFÉRICA. ....	31
3.3 ANEMÓMETRO.....	35
3.4 DETECTOR DE LLUVIA .....	38
3.5 MODULO WIFI.....	42

3.6 CMOS-TTL .....	46
3.7 PROTOTIPO.....	47
4. ANÁLISIS COMPARATIVO DE LOS COSTOS ENTRE LA SOLUCIÓN PLANTEADA Y LAS EXISTENTES EN EL MERCADO.....	48
4.1 ESTACIONES EXISTENTES.....	48
4.1.1 Estación meteorológica WatchDog WD 2700 .....	48
4.1.2 Estación meteorológica WD-2900. ....	49
4.1.3 Estación meteorológica WMR300 de Oregon Scientific.....	50
4.1.4 Estación meteo Davis Vantage Pro 2 Aktiv Plus 6163EU.....	51
4.2 COMPARACIÓN DE COSTOS. ....	52
5. PRUEBAS DE VALIDACIÓN DEL PROTOTIPO .....	54
5.1 HORA DEL SERVIDOR. ....	54
5.2 PRUEBA DE CONEXIÓN WIFI.....	54
5.3 PRUEBA PRESIÓN ATMOSFÉRICA.....	55
5.4 PRUEBA CAMBIOS DE TEMPERATURA .....	56
5.5 PRUEBA CAMBIOS DE HUMEDAD. ....	57
5.6 PRUEBA DETECCIÓN DE LLUVIA .....	58
5.7 PRUEBA VELOCIDAD DEL VIENTO.....	59
CONCLUSIONES .....	61
REFERENCIAS .....	62

## LISTA DE TABLAS

Tabla 1. Efectos esperados del cambio climático.(Continuación).....	18
Tabla 2. Escala de Beaufort.....	26
Tabla 3. Especificaciones técnicas .....	27
Tabla 4. Especificaciones sensor BMP180.....	32
Tabla 5. Comparación de costos entre tecnologías existentes y prototipo. ....	52



## LISTA DE FIGURAS

Figura 1. Sensor Dht11.....	28
Figura 2. Diagrama de conexión Dht11 a Raspberry Pi.....	28
Figura 3. Conexión Física Dht11 a Raspberry Pi.....	29
Figura 4. Esquema del BMP180 .....	32
Figura 5. Conexión del sensor de presión atmosférica. ....	33
Figura 6. Diseño de anemómetro.....	36
Figura 7. Módulo de efecto hall.....	37
Figura 8. Conexión física del módulo de efecto hall.....	38
Figura 9. Anemómetro completo.....	38
Figura 10. Placa de baquelita. ....	39
Figura 11. Controlador detector de lluvia. ....	40
Figura 12. Datos obtenidos por el detector de lluvia visualizados por consola. ....	41
Figura 13. Conexión de detector de lluvia en Raspberry Pi. ....	42
Figura 14. Instalación de controlador 8188eu.....	43
Figura 15. Ejecución del comando <code>lsusb</code> .....	44
Figura 16. Ejecución del comando <code>lsmod</code> .....	44
Figura 17. Ejecución del comando <code>ifconfig</code> .....	45
Figura 18. Conexión del módulo WIFI en Raspberry Pi. ....	45
Figura 19. Chip 74AHCT541.....	46
Figura 20. Diseño final del Prototipo. ....	47
Figura 21. Estación meteorológica WatchDog WD 2700 .....	49

Figura 22. Estación meteorológica WD-2900 .....	49
Figura 23. Estación WMR300 .....	50
Figura 24. Base receptora. ....	51
Figura 25. Estación base. ....	52
Figura 26. Prueba sensor BMP180.....	56
Figura 27. Prueba cambios de temperatura.....	57
.....	57
Figura 29. Prueba de medición de humedad. ....	58
Figura 29. Prueba sensor YL-83. ....	59
Figura 30. Prueba medición del viento.....	60

## LISTA DE ANEXOS

Anexo A. Programa para la lectura de temperatura y humedad. ....	66
Anexo B. Código del anemómetro.. ....	67
Anexo C. Código implementado para el sensor YL-83. ....	69
Anexo D. Aplicación Web. ....	71

## INTRODUCCIÓN

Las estaciones meteorológicas se vienen construyendo desde hace mucho tiempo dado el interés y la utilidad de conocer los fenómenos climáticos. Es importante la medición adecuada de diferentes variables físicas como temperatura, humedad, velocidad de viento, lluvia etc. y para esto se usan instrumentos como termómetros, barómetros y anemómetros, que en conjunto y en sus comienzos conformaron las estaciones meteorológicas de tipo analógico. Debido a que el avance tecnológico en los últimos tiempos, ha brindado la posibilidad de mejoras significativas en todos los campos de la ciencia incluida la meteorología, se ha logrado el perfeccionamiento de estos instrumentos de medición, como también se ha hecho más eficiente el procesamiento de la información, mejorando la proyección y predicción del cambio climático.

El cambio climático es una realidad que se ve reflejado entre otros, en la frecuencia de las inundaciones en algunas regiones, en las temperaturas elevadas en invierno, o simplemente en el anticipado florecimiento de algunas plantas, como también ya es comúnmente escuchado el nombre de fenómenos como el Efecto Invernadero, el Fenómeno del Niño y el Calentamiento Global, que son consecuencia de este cambio y los cuales repercuten sobre el desarrollo sostenible y el equilibrio de la vida.

Las estaciones meteorológicas analógicas son generalmente voluminosas y de alto costo, de allí la importancia de apoyar este campo de estudio mediante un dispositivo que permita obtener las mediciones de variables físicas como la temperatura, la humedad, la presión atmosférica, la velocidad del viento y la detección de lluvia, a partir de una investigación basada en la utilización de nuevas tecnologías de software y hardware libres, que aunada al enfoque del internet de las cosas y la reducción de costos, dará lugar a un modelo de estación meteorológica capaz de efectuar recolección de las variables medidas y el procesamiento de datos, permitiendo el acceso de su información remotamente.

Este prototipo de estación puede ser de mucha utilidad para futuros desarrollos, ya no solamente en el ámbito de la adquisición y transmisión de variables atmosféricas, sino de utilidad en los temas de predicción, la mitigación, y la adaptación a los fenómenos derivados del Cambio Climático.

## 1. MARCO REFERENCIAL

### 1.1 ANTECEDENTES

#### 1.1.1 Estaciones meteorológicas que usan hardware libre en el mundo.

Existen numerosos proyectos que utilizan plataformas de hardware libre en el mundo, y que han abierto diferentes posibilidades de innovación en la educación, la electrónica y las telecomunicaciones, el entretenimiento y el ocio, la biología, la robótica, la medicina, entre otros.

Las placas más exitosas y usadas en los últimos años han sido Arduino, la cual fue la promotora de proyectos de hardware libre gracias a su facilidad de manejo y posibilidades de interconexión, la placa Raspberry Pi creada con fines educativos y que ha tomado gran fuerza al ser prácticamente un mini computador con su sistema operativo Raspbian y recientemente la aparición de placas como Udo, PcDuino y otras más que son placas con más potencia y posibilidades que Raspberry Pi, aunque más costosas.

Los proyectos que los usuarios diseñan y ensamblan normalmente son compartidos a través de los foros y páginas web de las plataformas de hardware libre, por ejemplo en el portal de *Raspberry Pi Foundation*<sup>1</sup> la información es subida por el desarrollador y es libre de decidir hasta qué punto la comparte.

Con respecto a estaciones meteorológicas utilizando hardware libre hay varias iniciativas, pero solo llegan a ser configuraciones que miden temperatura, presión atmosférica y humedad, parámetros sencillos pero que dejan la posibilidad de desarrollar productos más complejos a partir de estas iniciativas, ya que la filosofía de hardware libre y de los mismos usuarios, es permitir, continuar o mejorar un desarrollo previo.

La estación meteorológica más desarrollada ha sido un proyecto Inglés, conocido como AirP<sup>2</sup>i, que es una estación capaz de registrar datos sobre la calidad del aire y las condiciones meteorológicas, construida con una placa Raspberry Pi. Desarrollado por Alyssa Dayan y Tom Hartley, estudiantes del Westminster School de Londres y ganador de la competición PA Consulting Raspberry Pi que se realizó a principios del año 2013.

Otras iniciativas y proyectos se describen a continuación:

---

<sup>1</sup> RASPBERRY PI FONDATION. About us of raspberry pi. [en línea:]< <http://www.raspberrypi.org>> [citado el 18 de Febrero de 2015]

<sup>2</sup> AIRPI, The Raspberry Pi Wather Station. [en línea:]< <http://airpi.es>><http://www.raspberrypi.org>> [citado el 18 de Febrero de 2015]

Carletto, J. *et al*<sup>3</sup> en su artículo “Datos climáticos on line con software libre”, describe la implementación de software libre para la adquisición y puesta on-line de datos climáticos, provenientes de una central meteorológica de bajo costo. Se describe las características y configuración del software utilizado CUMULUS para adquisición de datos, y la instalación de un servidor web para la puesta on-line de los datos.

En sus conclusiones indican la implementación de un software gratuito logrando altas prestaciones en la presentación de los mismos, brindando una solución económica para poder publicar en tiempo real los datos climáticos del lugar. Además se tiene la propuesta implementada por Schwartzman *et al*<sup>4</sup> en su artículo para la Revista Iberoamericana de Ingeniería Industrial “Sistema de comunicación y transmisión de datos desde estaciones meteorológicas” la cual muestra un desarrollo en lenguaje.

Berenice Montalvo<sup>5</sup>, afirma que una estación meteorológica es importante ya que proporciona información relevante para tomar decisiones y en un momento determinado acciones de forma certera, además permite la aplicación de varias asignaturas como Física, Máquinas eléctricas, Microprocesadores, Programación, Electricidad y Magnetismo, Teoremas de circuitos, Sistema en tiempo real, Lenguajes de internet y Red.

En su trabajo desarrolló pruebas obteniendo resultados para la comprobación de su proyecto; realizó una comparación entre las mediciones representativas generadas por el prototipo y las mediciones de un servicio meteorológico llamado Open WeatherMap p. 52 la cual demostró una similitud entre las mediciones del prototipo y las del servicio meteorológico tomado como referencia.

Las deducciones de su trabajo mostraron la implementación y construcción de un software que interactúa y registra la información de una forma exitosa, adicional por medio de un sitio web se generó el monitoreo e interpretación de manera remota de las variables físicas medidas por el dispositivo.

Carlos Gómez Sergio Velásquez,<sup>6</sup> en su paper “MONITOREO DE VARIABLES ANALOGICAS USANDO RASPBERRY PI” describen el proceso de monitoreo de

---

<sup>3</sup> CARLETTTO. J. et al. Datos climáticos on line con software libre. Universidad Nacional de San Luis. Argentina. 2007.

<sup>4</sup> SCHVARTZMAN, David et al. Sistema de comunicación y transmisión de datos desde estaciones metereologicas.En: Iberoamerican Journal of Industrial Engineering. Brasil. 2012, Vol. 4.

<sup>5</sup> MONTALVO, Berenice. Prototipo didáctico de una estación meteorológica monitoreada a distancia. Instituto Politécnico Nacional. México. 2013.

<sup>6</sup> VELÁSQUEZ, Sergio et al. Monitoreo de variables analógicas usando Raspberry pi. En: Universidad, ciencia y tecnología. [en línea:] N°69 (Diciembre de 2013) <<http://www.poz.unexpo.edu.ve/postgrado/uct/revista/index.php/uct/article/viewFile/267/218>> [citado el 18 de Febrero de 2015]

variables analógicas de forma “In situ” el cual presenta grandes ventajas debido a que la forma de procesamiento de datos se hace directamente con un computador el cual permite realizar distintas tareas a la vez, usando una tarjeta Raspberry PI cuyas características tienen un gran desempeño, ya que presenta diversas herramientas como por ejemplo el módulo GPIO (*General Purpose Input/Output*, Entrada/Salida de Propósito General) que permite desarrollar distintas acciones que dan lugar a una infinidad de aplicaciones.

En su desarrollo, este proyecto se basó en dos componentes importantes, el hardware conformado por los circuitos integrados, componentes pasivos, periféricos adicionales y el software. Para cumplir con su objetivo se requirió una etapa previa de adquisición de datos y el uso de un periférico que permitiera la obtención en imagen del proceso de forma constante.

Se realizó una prueba de acceso remoto mediante VNC (Virtual Network Computing, Computación Virtual en Red), el cual requerida de un servidor VNC en la tarjeta Raspberry, la cual funcionó satisfactoriamente introduciendo la dirección IP asignada al servidor y su contraseña. Dedicaron un capítulo entero para describir el proceso con el cual se calibraron los valores de adquisición obtenidos por los sensores mediante potenciómetros y describieron las pruebas realizadas y cuyos resultados fueron satisfactorios debido a la precisión y estabilidad del sistema implementado.

En sus conclusiones se justificó el uso una tarjeta Raspberry Pi ya que usaron varios periféricos para la adquisición de datos los cuales todos eran compatibles con esta tarjeta. Además, esta facilitó la conexión y comunicación por medio de un acceso remoto por conexión TCP/IP, resaltando que la herramienta GPIO que posee la tarjeta es una gran herramienta puesto que permite desarrollar varias acciones a la vez. SimpleCV fue el software que se implementó ya que es muy eficiente para obtener imágenes provenientes de cámaras Web USB.

Un proyecto previo a la obtención del título de físico realizado por Vascones Freddy Gustavo<sup>7</sup> de El politécnico nacional, describe un sistema de recolección de datos, dirigido al estudio climático en parámetros meteorológicos.

El sistema conformado por un hardware que toma las variables físicas como temperatura, humedad relativa, luminosidad y cantidad de lluvia. Además, este tiene otra parte más que se encarga de la transmisión de los datos por medio de internet.

---

<sup>7</sup> VÁSCONEZ ALBÁN, Freddy Gustavo. Diseño e implementación de un prototipo de estación de medición de parámetros meteorológicos para su incorporación en una red extendida de monitorización ambiental. Tesis para optar por el título de Ingeniero Electrónico. Quito. Escuela Politécnica Nacional. 2009.

El sistema de recolección de variables análogas como lo son las magnitudes físicas a medir, se les realiza un debido muestreo basado en el principio fundamental de Nyquist para así puedan los conversores, análogo a digital, puedan entregar muestras lo suficientemente eficientes para tener en cuenta de manera segura las mediciones. Un conversor de estos mantiene la información almacenada momentáneamente, para lo cual se decide utilizar un micro controlador, el cual además de haber fijado la frecuencia de muestro, recibirá toda la información y actuara según ella.

El controlador fue un PIC16f870 del fabricante Microchip el cual consta de 5 canales de conversión analógica-digital de 10 bits, dos periféricos, comunicación serial, tres temporizadores y un sistema WDT(Watch Dog Timer). Se usó un sistema Telos que es un módulo inalámbrico de ultra bajo consumo energético, diseñado para usarse en redes de sensorica conformado por un microcontrolador, un radio digital, una memoria flash, un microcontrolador para conexión USB, una antena, un sensor de temperatura y humedad relativa, un foto diodo sensor de radiación fotosintéticamente activa y dos puertos de expansión.

Concluyeron que este sistema fue muy versátil para la medición de variables físicas, gracias a que los sistemas Telos(sistema telefónico IP talkshow multilínea), pueden acoplarse con facilidad con distintos tipos de sensores mostrando que es posible integrar sensores que entregan tanto señales análogas de diferencia de potencial, como señales de intensidad de corriente eléctrica, como lo son las señales digitales. Gracias a esa implementación se dio a conocer que es posible la creación de un sistema de recolección de datos confiable y robusto que puede comunicarse fácilmente a internet y del cual se puede obtener información en tiempo real desde cualquier lugar del mundo.

También es importante resaltar que de este proyecto se pudo obtener información confiable a partir del análisis de los resultados obtenidos, dando campo para realizar más alcances a futuro como por ejemplo la entrega de reportes vía Email o anuncios en la Web sobre el cambio climático. Sin embargo aún se puede tener más proyección queriendo tener un control remoto del sistema para especificar el análisis de datos en un tiempo puntual.

**1.1.2 Estaciones meteorológicas utilizando hardware libre en Colombia.**  
Algunos antecedentes orientados al uso de tecnologías de información y comunicaciones para su contribución con el cambio climático global son:



La propuesta que hace Mendoza<sup>8</sup> en su tesis para optar al título de ingeniero electrónico “Metodología de diseño para la implementar un sistema de adquisición de datos meteorológicos para aeropuertos” en la cual expone la metodología para el desarrollo de diseño de un sistema de adquisición de datos meteorológicos, haciendo una propuesta de los elementos a utilizar lo cual le permite estudiarlos enumerando las características más importantes que le facilita desarrollar el modelo del circuito con cada uno de los componentes a utilizar con base en el principio de funcionamiento del sistema de adquisición de datos. Dentro de las conclusiones de su trabajo se encuentra el diseño un prototipo que posee características necesarias en los aeropuertos para la medición de variables climáticas e indica que se debe tener un criterio meteorológico y de dispersión atmosférica de la región donde se desarrolla el proyecto con el objetivo final de aumentar la seguridad vial comprobando la climatología completa de las rutas por las cuales se va a viajar.

López Carvajal B.<sup>9</sup> en su tesis para optar por el título de ingeniero electrónico “Diseño de hardware y software para una estación meteorológica integrando un pluviómetro, anemómetro y veleta para utilizarlos como instrumentos de medición” expone el diseño de una estación meteorológica integrando elementos como anemómetro, veleta y pluviómetro basado en software y hardware libre mostrando como resultado variables físicas como lo son la precipitación, velocidad y dirección del viento y aplicar los resultados obtenidos según la necesidad. Adicional implementa en su desarrollo el Arduino Nano® es una pequeña board, completa y fácil de usar y se basa en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x). Tiene más o menos la misma funcionalidad de la Arduino Duemilanove, pero en un paquete diferente. Le falta sólo una toma de alimentación de CC, y funciona con un cable mini-B USB en lugar de una estándar. El Arduino Nano® fue diseñado y está siendo producido por Gravitech.

En sus conclusiones afirma que el uso de nuevas tecnologías facilita la obtención de la información y el procesamiento de los datos; el Arduino nano fue de vital importancia en la implementación del proyecto no solo por su eficiencia y facilidad de manejo sino por el lenguaje de programación que propone que fue clave para la integración de los distintos módulos de la estación meteorológica. Entre los casos destacables de universidades que promueven el hardware libre, sitios web y proyectos de investigación están:

---

<sup>8</sup> MENDOZA ROZO, Edgar Yesid. Metodología de diseño para implementar un sistema de adquisición de datos meteorológicos para aeropuertos. Tesis para optar por el título de Ingeniero Mecánico. Pamplona. Universidad de Pamplona. 2006. 61 pag.

<sup>9</sup> LOPEZ CARVAJAL, Bryan Humberto. Diseño de Hardware y software para una estación meteorológica integrando un pluviómetro, anemómetro y veleta para utilizarlos como instrumentos de medición. Tesis de pregrado para optar por el título de Ingeniero Electrónico. Colombia. Universidad San Buenaventura Seccional de Medellín. 2013. 15 pág.

1.1.2.1 Web Frambuesa Pi Colombia. Página y comunidad web que explora el uso del Raspberry Pi, y busca promover a desarrolladores interesados en trabajar con esta herramienta de hardware libre, además comparte proyectos fáciles de implementar que se ensamblaron en otras partes del planeta y avances colombianos con el uso de la plataforma. La página de la comunidad es: <http://www.frambuesapi.co/><sup>10</sup>

1.1.2.2 Comité de Software Libre de Cali. Se definen como una meta-grupo que reúne a los diferentes grupos relacionados con el software/hardware/cultura libre de la región, con el propósito de difundir las tecnologías libres, compartir conocimiento y motivar su uso en las organizaciones, e invita a las universidades y estudiantes a ser parte de la iniciativa y desarrollar proyectos de hardware libre, además organiza talleres, charlas y eventos informativos de las tecnologías de software y hardware libre. Se puede acceder a la información en el sitio web: <http://wiki.softwarefreedomday.org/2013/Colombia/Cali/CSLC><sup>11</sup>

1.1.2.3 Semillero de investigación SIAN Open Hard. En la Universidad Tecnológica de Pereira existe el semillero de investigación SIAN Open Hard (Semillero de Investigación Android y Hardware Libre), está formado por grupo de jóvenes se encuentran desarrollando proyectos con Android y Arduino. La página de la universidad ofrece un link al blog del semillero, e informa sobre las actividades que realiza el mismo y es: <http://blog.utp.edu.co/alejandropinto/tag/semillero-de-investigacion/><sup>12</sup>

**1.1.3 El cambio climático en Colombia.** El cambio climático puede presentar dificultades en los niveles del desarrollo humano en países que están en este proceso y que además poseen comunidades pobres y vulnerables, es por eso que se debe tomar conciencia en los impactos que puede llegar a tener el cambio climático en nuestro país debido a que su ubicación geográfica presenta zonas muy expuestas a la inundación y también presenta suelos inestables con escases de agua debido a las cordilleras, haciendo que Colombia sea un país que en todo momento está expuesto a desastres naturales asociados al clima.<sup>13</sup>

---

<sup>10</sup> RASPBERRY PI. Frambuesa Pi Colombia. [en línea:]< <http://www.frambuesapi.co/> > [citado el 18 de Febrero de 2015]

<sup>11</sup> SOFTWARE FREEDOM DAY. Cslc. [En línea:]<<http://wiki.softwarefreedomday.org/2013/Colombia/Cali/CSLC>> [citado el 25 de Febrero de 2015]

<sup>12</sup> PINTO, Alejandro. Emprendimiento Digital y TIC. [en línea:]<<http://blog.utp.edu.co/alejandropinto/tag/semillero-de-investigacion/>> [citado el 16 de Febrero de 2015]

<sup>13</sup> Programa de las Naciones Unidas para el Desarrollo en Colombia- PNUD, «El cambio climático en Colombia y en el sistema de las Naciones unidas,» 5 Enero 2010. [En línea]. Available: [http://www.pnud.org.co/img\\_upload/61626461626434343535373737353535/Brochure%20resumen%20Proyecto.pdf](http://www.pnud.org.co/img_upload/61626461626434343535373737353535/Brochure%20resumen%20Proyecto.pdf). [Último acceso: 16 02 2015].

La Segunda Comunicación Nacional de Cambio Climático de Colombia presenta que el aumento de temperatura media fue de 2 a 4 grados al 2070 y que se reducirán, en algunas regiones, las precipitaciones hasta en un 30%. Todo esto genera un impacto en muchos sectores y dificulta la lucha contra la protección de comunidades vulnerables. En la Tabla 1 se pueden ver los efectos esperados debido al cambio climático.

Tabla 1. Efectos esperados del cambio climático.

Algunos de los efectos esperados del cambio climático	
Salud	Aumento de la incidencia de las enfermedades transmitidas por vectores (malaria y dengue). Las regiones andinas son las más propensas a estas nuevas epidemias por ser zonas de malaria inestable pero también indirectamente del deterioro de los recursos hídricos y de las condiciones de habitabilidad.
Sector agropecuario	Gran parte de los agros ecosistemas del país son vulnerables a los efectos de la aridización, la erosión de suelos, la desertificación y a los cambios en el régimen hidrológico. También se estima un mayor riesgo de inundaciones en cultivos y de otros eventos naturales que afectan la producción agrícola (vendavales, granizadas, etc...).
Recursos hídricos	Se proyectan aumentos de la escorrentía en las regiones costeras, en los llanos orientales y en los departamentos donde en las últimas décadas hubo inundaciones y deslizamientos. En contraste se prevé una disminución de la escorrentía en la región andina y el norte del país, que puede causar problemas en el suministro de agua y déficit en los embalses, lo cual produce la disminución de hidroenergía.
Sistemas costeros	Con la elevación prevista del nivel del mar millones de habitantes quedarían expuestos a inundaciones en las zonas costeras, como así también los asentamientos industriales, la infraestructura e instalaciones turísticas y los cultivos. También son vulnerables las fuentes de agua por una mayor intrusión salina.
Ecosistemas	Preocupa la reducción del área de nevados y paramos y por lo tanto, de sus servicios ambientales. Los corales pueden sufrir por el aumento de la temperatura media del mar, y así afectar la biodiversidad, y los recursos pesqueros asociados. Los impactos sobre los bosques pueden ser considerables pero existe todavía la incertidumbre sobre la resiliencia de éstos.
Vivienda y asentamientos	La infraestructura en todo el país y los asentamientos precarios y en zonas de riesgo pueden verse afectados por la mayor frecuencia de eventos extremos (en especial inundaciones, lluvias fuertes, tormentas tropicales, vendavales y deslizamientos), lo que deteriora más aún en condiciones de habitabilidad y la calidad de vida de poblaciones desplazadas y pobres.

Fuente: Ministerio de asuntos exteriores y de cooperación. El cambio climático en Colombia y en el sistema de las Naciones Unidas.

Como es mencionado en el documento “El cambio climático en Colombia y en el sistema de las Naciones Unidas” presentado por el programa de las Naciones Unidas para el desarrollo en Colombia<sup>14</sup>, el interés sobre el cambio climático en el mundo aumenta, el aprender a disminuir la vulnerabilidad, mejorar la capacidad de adaptación, disminuir las emisiones de gases de efecto invernadero y sobre todo mejorar las decisiones teniendo en cuenta el contexto de cambio climático, serán las herramientas que ayudaran a que la población pueda mejorar su débil condición frente a los desastres que pueden ocurrir debido a esto.

<sup>14</sup> Programa de las Naciones Unidas para el Desarrollo en Colombia- PNUD, «El cambio climático en Colombia y en el sistema de las Naciones Unidas», 5 Enero 2010. [En línea]. Available: [http://www.pnud.org.co/img\\_upload/61626461626434343535373737353535/Brochure%20resumen%20Proyecto.pdf](http://www.pnud.org.co/img_upload/61626461626434343535373737353535/Brochure%20resumen%20Proyecto.pdf). [Último acceso: 16 02 2015].

La meteorología es, como lo menciona M. Cote, P. Martin, J. Gonzales I. y A. Cardona<sup>15</sup>, la ciencia que se encarga de estudiar las propiedades de la atmosfera y sus fenómenos en relación con la superficie de la tierra y el mar, y también con el tiempo atmosférico. El movimiento de la atmosfera, la radiación solar, los procesos termodinámicos que dan origen a la formación de las nubes y las diferentes razones de la precipitación son algunos de los enfoques de la meteorología basados en el conocimiento de una serie de magnitudes como la temperatura, la presión atmosférica y la humedad que varían dependiendo del momento y el espacio.

Como lo mencionan R. M. Rodriguez, A. Capa, A. Portela, F. Martín, J. Quirantes y F. Rey<sup>16</sup>, preguntarse por qué a veces se ven noticias de que se han venido presentando más inundaciones frecuentemente, o porque las temperaturas en invierno son extrañamente más elevadas que el año anterior, y que quizá porque el cambio de estación se dio más temprano de lo normal. Estas preguntas son aquellos indicios de que se han ido presentando cambios climáticos causando fenómenos muy escuchados en la actualidad como el efecto invernadero que consiste en que el desgaste de la atmosfera que tiene como objetivo dejar pasar luz solar haciendo que se retenga más calor, pero pues como existe más desgaste de la atmosfera, el calor del sol revotaría en la superficie de la tierra y se perdería, provocando que la temperatura de la tierra se disminuya al punto de que todo se congelaría.

**1.1.4 Transmisión de información.** Todo sistema de comunicaciones lo comprende un transmisor, un canal y un receptor. Cada uno de estos se compone de elementos que efectúan un proceso con el fin de establecer la buena comunicación.

El proceso para enviar una señal comienza por una señal de información que transmitirá música, video, datos, luego actúa una señal portadora de mayor potencia que no lleva información, si no en la cual se enlaza la señal de información para poder ser transmitida.

Luego una etapa de modulación en la que se modifican las señales con el propósito de enviar más información por el canal y brindarle más seguridad a la transmisión. Por último se amplifica la señal con el propósito de que el mensaje transmitido alcance su receptor con claridad.

---

<sup>15</sup> M. Cote, P. Martin, J. Gonzales I. y A. Cardona, «El cambio climatico en Colombia y en el sistema de las Naciones Unidas,» PNUD, New York, 2010.

<sup>16</sup> R. M. Rodriguez, A. Capa, A. Portela, F. Martín, J. Quirantes y F. Rey, «¿Que es la meteorología?,» de METEOROLOGÍA Y CLIMATOLOGIA, España, Fundación Española para la ciencia y la tecnología, 2004, pp. 612- 620.

El medio o canal por donde se genera la comunicación nos brinda las limitaciones de información que este puede alcanzar a transmitir, en este caso de las comunicaciones inalámbricas el medio es el aire por el que viajan ondas electromagnéticas que transportan la información.

El proceso de recepción de la señal comienza por un proceso de amplificación de la señal que puede llegar con una potencia muy baja debido a las pérdidas que puede presentar el medio por el cual se transmite. Luego se hace la etapa de demodulación en donde se transforma la señal codificada en una señal digital normal y en la parte final del proceso, un transductor traduce la señal recibida en la información original<sup>17</sup>.

Los sistemas de comunicación inalámbricos están clasificados desde el punto de referencia normativo sobre bandas reguladas y no reguladas. El Wifi funciona en las bandas no reguladas que está definido por el estándar IEEE 802.11 y que se creó con el propósito de brindar conectividad en redes de área local (LAN), en las que generalmente se usa la conexión a internet inalámbricamente. Ubicado en la banda de 2.4Ghz, tiene un alcance de metros y su velocidad de transmisión puede de ser de 11Mbps<sup>18</sup>.

---

<sup>17</sup> Organización de Estados Iberoamericanos - OEI, «el cambio climático: ¿Que es?,» 10 Febrero 2006. [En línea]. Available: [http://www.oei.es/decada/portadas/climate\\_change\\_youth\\_es.pdf](http://www.oei.es/decada/portadas/climate_change_youth_es.pdf). [Último acceso: 6 Marzo 2015].

<sup>18</sup> D. F. Valencia Z., Diseño y construcción de un sistema de adquisición de la temperatura, humedad, velocidad y dirección del viento para la evaluación del potencial micro-elico, Cali: Universidad de San Buenaventura, 2013.

## **2. IDENTIFICACIÓN DE VARIABLES FÍSICAS A MEDIR**

Esta investigación pretende analizar y realizar la adquisición de las variables físicas más comúnmente conocidas, pues son la base fundamental para un estudio meteorológico y las que permitirán entender más el funcionamiento de las nuevas tecnologías y sus alcances para esta o cualquier aplicación.

### **2.1 IDENTIFICACIÓN DE VARIABLES FÍSICAS A MEDIR**

Una estación meteorológica es un sistema conformado por sensores, medios de comunicación y procesamiento de información, cuyo propósito es medir y registrar cualquier tipo de variación física relacionada con el clima.

Las variables climáticas que son generalmente medidas por una estación meteorológica, son la temperatura, la humedad relativa, la presión atmosférica y la precipitación. El interés de medir otras variables dependiendo del propósito del estudio como lo es por ejemplo la medición de radiación solar por medio Pirómetro, la medición de las horas de luz que tiene cada día por medio Heliógrafo y hasta también la medición de la altura de las nubes por medio de un Nefobasímetro.

La estación meteorológica que se propone, estará ubicada en un terreno libre de obstrucciones, siendo de nuestro interés medir las variables físicas más comunes y crear un sistema con el cual se puedan obtener información de cada una de ellas, teniendo en cuenta siempre el propósito de crear un sistema preciso, fiable y sobre todo de bajo costo.

**2.1.1 Temperatura.** La temperatura es una magnitud escalar relacionada con la energía cinética asociada con los movimientos de las partículas de un sistema y que para poder ser medida nos basamos en las propiedades de la materia que al presentar alteraciones se pueden observar cambios en el volumen o el color de un objeto.

Siendo la temperatura una de las magnitudes más utilizadas, las unidades de medida son grados Fahrenheit, Celsius o Kelvin. Los grados centígrados están definidos entre los cambios de estado del agua, en el punto de ebullición (100 grados) y de fusión (0 grados). Los grados Fahrenheit tienen el punto de ebullición del agua en 212 grados y de fusión en 32 grados, en cambio los grados Kelvin tienen su punto de fusión del agua en 273,15 grados y el punto de ebullición en 373,15 grados.

La medición de esta variable se hace a través de una invención hecha por Galileo en 1593 llamado termómetro, el cual en su forma más básica y sencilla consiste en un tubo que tiene en su interior un fluido que puede ser alcohol o mercurio, ya

que como estos líquidos se expanden más que el vidrio, ante la presencia de aumentos de temperatura estos fluidos ascienden por el tubo<sup>19</sup>.

Una de las limitantes más importantes en el crecimiento natural de las plantas o de la vegetación en general, es la temperatura. Todos los tipos de plantas que existen tienen una temperatura específica en la cual se optimiza su crecimiento, y que si esta temperatura baja o sube, el crecimiento de la planta se detiene. Las hortalizas de estación fría crecen de la mejor forma en temperaturas entre los 15 y 18 grados centígrados.

**2.1.2 Humedad.** La cantidad de vapor de agua en el aire varía con el tiempo y el lugar, dependiendo de la temperatura el vapor no podrá superar su valor máximo de presión y por eso solo una cantidad de volumen de aire puede contener un valor máximo de vapor de agua en ella. La humedad máxima se encuentra a una determinada temperatura en un metro cubico de aire pero normalmente nunca se alcanza el nivel de saturación del aire suficiente para obtener ese valor máximo<sup>20</sup>.

La presión de saturación es la presión que hace el vapor sobre el aire que depende de la temperatura, entre mayor sea la temperatura, mayor es la cantidad de saturación y menor temperatura el aire puede almacenar menor cantidad de vapor de agua.

**2.1.2.1 Humedad absoluta.** Humedad absoluta es la relación entre el peso del agua y el volumen de aire, en otras palabras, al medir esta humedad se obtiene la cantidad de vapor que contiene el aire, y si se tiene la información sobre la temperatura en ese momento se podrá determinar si en este ambiente el aire es capaz de alojar más vapor.

**2.1.2.2 Humedad específica.** Es la misma humedad absoluta pero sus unidades de medición son aquellas que la hacen distinta, midiendo el vapor en gramos y el aire en kilogramos y se encuentra en estado gaseoso.

---

<sup>19</sup> K. C. Laudon y J. P. Laudon, Sistemas de información Gerencial, Reino Unido: Pearson Educacion, 2001.

<sup>20</sup> . 19, 20 Wikipedia, «Wikipedia la enciclopedia libre,» Fundación Wikimedia, Inc., Enero 1990. [En línea]. Available: <http://es.wikipedia.org/wiki/Temperatura>. [Último acceso: 2 Febrero 2015].

2.1.2.3 Humedad relativa “La cantidad de humedad en el aire, comparado con la que el aire puede mantener a esa temperatura”<sup>21</sup> esa es la humedad relativa. Por ejemplo “si la humedad es del 50% a 23 ° C, esto implicaría que el aire contiene 50% del nivel máximo de vapor de agua que podría mantener a 23 ° C. 100% de humedad relativa, indica que el aire está en la máxima saturación”<sup>22</sup>

La humedad está implicada directamente con el consumo de agua por parte de las plantas, dando lugar a tener un sistema de riego. Cuando la planta se encuentra en condiciones en las que la humedad es demasiado alta, su consumo de agua es tan bajo causando déficits en la absorción de nutrientes, que van a presentar deficiencias en la producción en un cultivo. El calcio y el potasio se absorben a través de las raíces, al presentarse este tipo de condición, se verá reflejada tanto en las hojas como en sus frutos, la falta de estos nutrientes, que como consecuencia pueden dar lugar a enfermedades como por ejemplo la Phythophthora, la cual permite observar quemadura en la hojas de las plantas debido a que ellas permanecen húmedas más tiempo de lo normal.

**2.1.3 Presión atmosférica.** La presión que ejerce la atmosfera sobre cualquier objeto ubicado en cualquier punto de la superficie, se le llama presión atmosférica. Esta presión atmosférica también se entiende como la cantidad de aire por encima de un punto hasta el punto más alto de la atmosfera, lo cual quiere decir que entre más alto sea el punto en la superficie con respecto al nivel del mar, menos presión atmosférica tiene este objeto<sup>23</sup>.

La presión atmosférica es de gran influencia en los fenómenos climatológicos. Como lo menciona el documento

Si la corriente de aire es fría, baja y hace subir la presión, contribuyendo a la formación de lo que se conoce como anticiclón térmico. En cambio, si el aire es tibio o caliente, se incrementa y promueve una bajada de la presión, lo que lleva a la formación de un ciclón térmico y provoca inestabilidad<sup>24</sup>

La presión barométrica es la misma presión atmosférica local con la corrección por la altitud geo posicional y este valor oscila entre el valor de la presión atmosférica normalizada. Uno de sus propósitos es calcular la altitud.

---

<sup>21</sup> Georgia State University, «Humedad Relativa,» HyperPhysics, [En línea]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbasees/kinetic/relhum.html>. [Último acceso: 20 Marzo 2015].

<sup>22</sup> International, «¿Qué es la "Humedad Relativa"?», [En línea]. Available: <http://www.yachtpaint.com/esp/diy/ask-the-experts/qu%C3%A9-es-la-humedad-relativa.aspx>. [Último acceso: 20 Marzo 2015].

<sup>23</sup> Pontificia Universidad Católica de Chile, «Clasificación termica,» 20 Diciembre 1961. [En línea]. Available: [http://www7.uc.cl/sw\\_educ/hort0498/HTML/p015.html](http://www7.uc.cl/sw_educ/hort0498/HTML/p015.html). [Último acceso: 4 Marzo 2015].

<sup>24</sup> Definicion.De, «Definición de Presión Atmosferica,» [En línea]. Available: <http://definicion.de/presion-atmosferica/>. [Último acceso: 1 abril 2015].



**2.1.4 Velocidad del viento.** La diferencia de temperaturas en distintos lugares geográficos produce el movimiento del aire que es definido como el viento. El viento se mueve de los centros de presión más alta a los de más baja y se define por la dirección en el plano horizontal o vertical. Las diferentes temperaturas hacen que el aire se mueva en muchas direcciones debido a los fluidos que tiene la atmósfera que comúnmente no están en reposo.

El aire es más frío en las capas altas de la atmósfera y allí pierde vapor de agua, bajando hacia la superficie terrestre, este aire ejerce presión sobre el aire que está debajo. Este aire frío que viene bajando, no produce precipitaciones porque es muy seco y como consecuencia es difícil que exista sobre esta superficie vegetación<sup>25</sup>.

La influencia del comportamiento del viento es muy importante en el sector agrícola, puesto que puede generar tanto beneficios como desastres en los cultivos, por ejemplo el movimiento del viento permite que las plantas mejoren su transpiración renovando el aire, como también si su movimiento es a grandes velocidades puede causar la pérdida de frutos causando daños mecánicos, pero estos daños son bastante tangibles, hay otros como por ejemplo que el viento elevado puede impedir el vuelo de insectos polinizadores provocando limitaciones en la producción.

El asado es un fenómeno en el que coinciden temperaturas altas con viento seco provocando que las raíces de la vegetación no tengan la capacidad de reponer las pérdidas producidas, es cuando se empieza a observar que las hojas de las plantas presentan quemaduras y caída de sus frutos, caso que es visto mucho en los frutales y en suelos bajo riego o suelos con reserva de agua<sup>26</sup>.

La precipitación de algún lugar depende de la presión atmosférica, además también los tornados son provocados por la baja presión atmosférica que hay en su interior, provocando la alta velocidad del viento. La rápida rotación del aire provoca que esta se convierta en forma de gotas de agua, provocada por el vapor de agua que hay en el aire, permitiendo que el aire sea visible al aparecer una especie de embudo de condensación.

Investigaciones han dado veracidad de que en un tornado las variaciones de temperatura, humedad y sobre todo de la presión atmosférica son bastante evidentes, cerca de Manchester se presentó la disminución de presión atmosférica

---

<sup>25</sup> Pontificia Universidad Católica de Chile, «Clasificación térmica,» 20 Diciembre 1961. [En línea]. Available: [http://www7.uc.cl/sw\\_educ/hort0498/HTML/p015.html](http://www7.uc.cl/sw_educ/hort0498/HTML/p015.html). [Último acceso: 4 Marzo 2015].

<sup>26</sup> S. e. a. VELÁSQUEZ, «Monitoreo de variables analógicas usando Raspberry pi,» Universidad, ciencia y tecnología, Diciembre 2013. [En línea]. Available: <http://www.poz.unexpo.edu.ve/postgrado/uct/revista/index.php/uct/article/viewFile/267/218>. [Último acceso: 18 Febrero 2015].

a 100mbar dando lugar a un tornado que llevo la presión atmosférica en su vértice hasta 850mbar<sup>27</sup>.

**2.1.5. La lluvia.** La detección de lluvia es fundamental para el estudio hidrológico regional pues es usado para cuantificar los recursos hídricos en lugares en los que el único acceso de agua es a través de este medio, pero también es usado en la previsión del clima, el diseño de obras públicas, estudio de erosión, etc.

La lluvia, la nieve o el granizo, se identifican a través del mismo fenómeno llamado precipitación. Su magnitud aumenta cuando grandes cantidades de aire presentan una disminución en la temperatura y un ascenso elevado, esto hace que la presión atmosférica disminuya y este aire se expanda. Este aire pasa por procesos de condensación, proceso adiabático y gradiente.<sup>28</sup>

La intensidad de la precipitación es igual a la precipitación en relación con el tiempo, para esto son usados recipientes de paredes rectas, en los que se mide la cantidad de agua recogida en periodos de tiempo de un día. Sin embargo en nuestro caso, aun no se desea obtener cuantos milímetros de agua se recolectan en un día, o cuanta intensidad de precipitación se logra conseguir en una hora.

El propósito de nuestro sistema es lograr ser lo más eficiente en todos los aspectos, y uno de ellos también es el ahorro energético. No todo el tiempo está lloviendo, pero ¿si todo el tiempo se estará midiendo? Se desea primero detectar si está lloviendo o no para indicarle al sistema cuando hacer una medición y también lograr obtener una recolección de esta medición solo en estos momentos.

El estudio de esta variable física es bastante importante debido a que fenómenos como “el Niño” que consiste en el calentamiento anormal de las aguas del Pacifico, que se trasladan de Oeste a Este impulsadas por una débil corriente de origen desconocido, algún día pueda ser conocido, y también poder ser tratado al punto de que algún día se obtenga una solución a este fenómeno.<sup>29</sup>

En la tabla 2 podemos encontrar la clasificación del viento, dependiendo su velocidad y efectos causados en alta mar. Esta tabla también es llamada como escala de Beaufort.

---

<sup>27</sup> F. G. VÁSCONEZ ALBÁN, «Diseño e implementación de un prototipo de estación de medición de parámetros meteorológicos para su incorporación en una red extendida de monitorización ambiental,» Escuela Politecnica Nacional, Quito, 2009.

<sup>28</sup> Universidad Autónoma Chipango, «Precipitación,» [En línea]. Available: [http://www.chapingo.mx/irrigacion/planest/documentos/apuntes/hidrologia\\_sup/UNIDAD4-PRECIPITACION.pdf](http://www.chapingo.mx/irrigacion/planest/documentos/apuntes/hidrologia_sup/UNIDAD4-PRECIPITACION.pdf). [Último acceso: 19 Abril 2015].

<sup>29</sup> M. C. Carpio, «Precipitación,» [En línea]. Available: <http://www.olivacordobesa.es/Precipitaci%F3n.pdf>. [Último acceso: 15 Abril 2015].

Tabla 2. Escala de Beaufort

ESCALA ESCALA BEAUFORT (VIENTO) DOUGLAS (MAR)					
Grado	Denominación	Velocidad	Descripción	Nombre	Altura de Ola en Mts.
		nudos	Grado		
0	Calma	<1	Mar como un espejo. 0	CALMA	0
1	Ventolina	1-3	Rizos como escamas de pescado, pero sin espuma.  Pequeñas olas, crestas de apariencia vítrea, sin romperse. 1	RIZADA	0-0,25
2	Flojito (Brisa muy débil)	4-6			
3	Flojo (Brisa débil)	7-10	Pequeñas olas, crestas rompientes, espuma de aspecto vítreo aislados vellones de 2 espuma..	MAREJADILLA	0,25- 0.5
4	Bonacible (Brisa moderada)	11-16	Olas un poco largas. Numerosos 3 borreguillos.	MAREJADA	0,5 - 1,25
5	Fresquito (Brisa fresca)	17-21	Olas moderadas y alargadas. Gran abundancia de 4 borreguillos y eventualmente algunos rociones.	FUERTE MAREJADA	1,25 - 2,5
6	Fresco (Brisa fuerte)	22-27	Comienza la formación de olas grandes. Las crestas de espuma blanca se ven por doquier. Aumentan los rociones y la 5 navegación es peligrosa para embarcaciones menores.	GRUESA	2,5 - 4
7	Frescachón (Viento fuerte)	28-33	La espuma es arrastrada en dirección del viento. La mar es 6 gruesa.	MUY GRUESA	4-6
8	Temporal (Viento duro)	34-40	Olas altas con rompientes. La espuma es arrastrada en nubes blancas.  Olas muy gruesas. La espuma es arrastrada en capas espesas. La mar empieza a rugir. Los rociones dificultan la visibilidad.	ARBOLADA	6-9
9	Temporal Fuerte (Muy duro)	41-47			
10	Temporal Duro (Temporal)	48-55	Olas muy gruesas con crestas empenachadas. La superficie de la mar parece blanca. Visibilidad reducida. La mar ruge. 8  Olas excepcionalmente grandes (los buques de mediano tonelaje se pierden de vista). Mar completamente blanca. Visibilidad muy reducida.	MONTAÑOSA	9 - 14
11	Temporal Muy Duro (Borrasca)	56-63			
12	Temporal Huracanado (Huracán)	64-71 >	El aire está lleno de espuma y de rociones. La visibilidad es 9 casi nula.	ENORME	> 14

Fuente: NauticaTresCantos Escalas de Beaufort-Douglas.

### 3. DISEÑO DEL SISTEMA DE ADQUISICIÓN, PROCESAMIENTO Y TRANSMISIÓN DE LAS VARIABLES FÍSICAS SENSADAS

#### 3.1 SENSOR DE HUMEDAD Y TEMPERATURA

Se investigó y se encontró una gran variedad de sensores que permiten la medición de la temperatura y de la humedad de forma separada, como lo son el LM35 de temperatura o el DHT11 que mide temperatura y humedad a la vez, teniendo siempre en cuenta que la cantidad de hardware que usaría para conformar la estación meteorológica debería ser la más mínima posible pues esto reduce costos y mejora la operatividad y portabilidad del sistema. También se tuvo en cuenta que para desarrollar un buen sistema de adquisición de datos se requería sensores con una buena precisión, lo cual aumentaba los costos pero daba mayor fiabilidad en los datos, el sensor elegido fue el sensor DHT11.

DHT11 es un sensor que proporciona una salida de datos digital. Entre sus ventajas podemos mencionar el bajo costo y el despliegue de datos digitales.

Este es un sensor multifuncional que proporciona información de temperatura y humedad relativa en el mismo tiempo. Proporciona lecturas fiables cuando la condición ambiente de humedad entre 20% HR y 90% de HR, y condición de temperatura entre 0 ° C y 50 ° C, que cubre las necesidades en la mayoría de aplicaciones del hogar y diarias que no contienen condiciones extremas, las especificaciones se describen en la tabla 3. El sensor se puede observar en la figura 1.

Tabla 3. Especificaciones técnicas

<b>Fuente de Alimentación</b>
3.3 - 5.5 V DC
<b>Señal de salida</b>
Señal digital a través de MaxDetect Bus
<b>Rango de funcionamiento</b>
Humedad relativa 20 a 90%
Temperatura 0 a 50 grados C
<b>Precisión</b>
Humedad +-5%RH
Temperatura +- 2 grado C
<b>Resolución o sensibilidad</b>
Humedad 1%
Temperatura 1 grado C

Figura 1. Sensor Dht11.

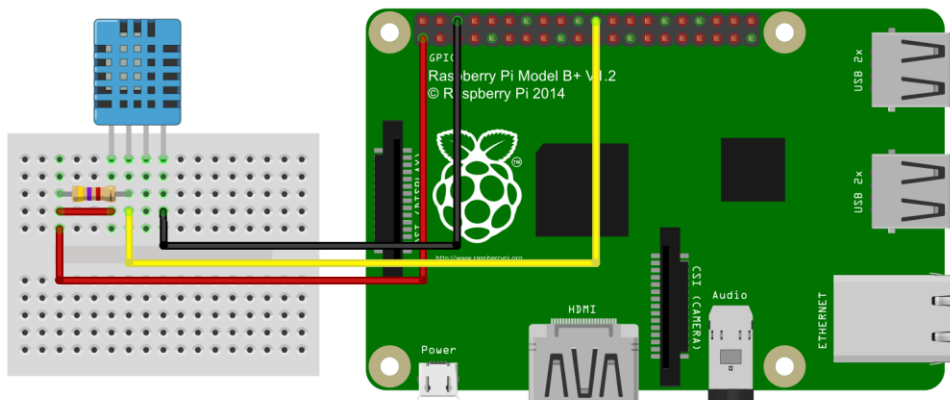


Fuente: Dirux.com. Sensor de humedad y temperatura.<sup>30</sup>

### Esquema conexión DHT11 físico a la RASPBERRY PI:

A continuación se presenta el diagrama de conexión para integrar el sensor Dht11 con la tarjeta Raspberry Pi en la figura 2 y también se puede observar la conexión física en la figura 3.

Figura 2. Diagrama de conexión Dht11 a Raspberry Pi.

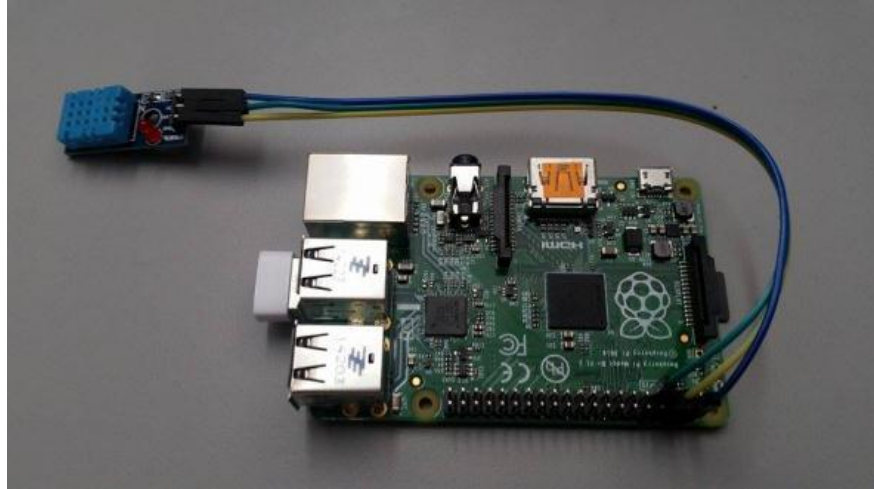


Fuente: Filipeflop. Monitoreando temperatura con Dht11 e Raspberry Pi.<sup>31</sup>

<sup>30</sup> Dirux.com, «Sensor de humedad y temperatura,» [En línea]. Available: <http://www.dirux.com/productos/electronica/dht11.php>. [Último acceso: 19 Abril 2015].

<sup>31</sup> Filipeflop, « Monitoreando temperatura con Dht11 e Raspberry Pi,» [En línea]. Available: <http://blog.filipeflop.com/embarcados/temperatura-umidade-dht11-com-raspberry-pi.html>. [Último acceso: 19 Abril 2015].

Figura 3. Conexión Física Dht11 a Raspberry Pi.



Fuente: Propia del autor.

También se puede añadir un condensador de 100 nF entre VDD y GND para filtrar la alimentación, aunque sin él también funcionará, dependiendo del encapsulado, se tendría que añadir resistencias y condensadores a la conexión.

### Montaje en la Raspberry PI.

Se conecta el sensor DHT11 a los puertos GPIO de la Raspberry de la siguiente manera:

RPi VCC (pin 1) -> DHT11 pin 1  
RPi GPIO4 (pin 7) -> DHT11 pin 2  
RPi GND (pin 6) -> DHT11 pin 4 (pin3 si solo tiene 3 pins)

Instalación de complementos.

- Compiladores para la programación en lenguaje C.
- Librería para la manipulación de los puertos digitales de la Raspberry Pi (WiringPi).

```
sudo apt-get install git-core build-essential
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

La instalación de servidor Apache se realizó usando los siguientes comandos:

```
apt-get install apache2
service apache2 start
```

Librería Dygraphs para la generación de la gráfica en la página web:

```
wget -P /var/www http://dygraphs.com/dygraph-combined.js
```

La creación del programa “*dth11.c*”, para acceder y leer la temperatura y humedad del sensor se puede encontrar en el Anexo A.

Compilación el programa con el siguiente comando.

```
gcc -o dth11 dth11.c -L /usr/local/lib -l wiringPi
```

Se ejecuta el programa con el siguiente comando si la compilación quedó correcta:

```
sudo ./dth11
```

Se necesita la creación de un fichero para que se guarde la información de forma automática, así que se procede a la creación del mismo con el siguiente comando:

```
cd ~
```

La creación del fichero se realiza con el siguiente comando:

```
nano dth11.sh
```

Código usado para usar el fichero:

```
#!/bin/bash
FECHA=$(date +%Y%m%d%H%M%S)
COMA=", "
TEMP=$(/home/pi/dth11)
echo "$FECHA$COMA$TEMP" >> /var/www/temp.log
```

Ahora se realiza la tarea programada o cron para que se ejecute cada minuto, para esto se usa el siguiente comando:

```
sudo crontab -e
```

Se añade la siguiente línea al final del cron:

```
* * * * * /home/pi/dth11.sh
```

Para la muestra de los resultados, se crea una página web en PHP cuyo código se puede visualizar en el Anexo D y que se alimenta a través de una base de datos Mysql que también es implementada en la misma tarjeta de desarrollo.

### **3.2 SENSOR DE PRESIÓN ATMOSFÉRICA.**

Guiados por estos nuevos complementos que se han desarrollado para poder realizar cualquier aplicación con Raspberry Pi y Arduino, nos encontramos con un sensor bastante apto para poder realizar las mediciones de la presión atmosférica teniendo un equilibrio entre el costo y su eficiencia en la adquisición de esta variable física, el sensor elegido es:

#### **Sensor BMP180 (Sensor Barométrico).**

##### **Especificaciones.**

Sensor de alta precisión con un rango de medida de entre 300 y 1100 hPa (Hecto Pascal) con un margen de error mínimo de tan sólo 0.03 hPa. Está basado en tecnología piezo-resistiva de alta eficiencia, linealidad y larga duración. El sensor tiene un rango de alimentación de entre 1,8V y 3,6 Vdc. Está diseñado para ser conectado directamente a un micro controlador mediante su interfaz I2C. Dispone de dos resistencias pull-up de 4,7k sobre el bus I2C.

Éste tipo de sensores pueden ser utilizados para calcular la altitud con bastante precisión, por lo que son muy útiles en UAV(Unmanned Aircraft Vehicle) conocidos popularmente como los drones.

Este modelo incluye un pequeño jumper (soldado) para poder desconectar las resistencias Pull-Up I2C. Las especificaciones también están descritas en la tabla 4, y su conexión física se puede visualizar en la figura 4.

##### **Características:**

- Interfaz de dos cables I<sup>2</sup>C
- Amplio rango de medición
- Alimentación: 1,8 - 3,6Vdc
- Consumo extremadamente pequeño
- Medición de bajo ruido
- Calibrado de fábrica
- Incluye medición de temperatura para compensación
- Muy pequeño y ligero
- Dimensiones: 16,5x16,5mm



Tabla 4. Especificaciones sensor BMP180.

Parámetro	Condición	Mínimo	Máximo	Unidades
Temperatura		-40	85	°C
Voltaje	Todos los pines	-0.3	+4.25	V
ESD Rango	HBM, R = 1.5kΩ, C = 100pF		±2	kV
Presión			10	hPa

Fuente: Bosch Invented for life. BMP180 Digital pressure sensor.<sup>32</sup>

Figura 4. Esquema del BMP180



Fuente: Del autor.

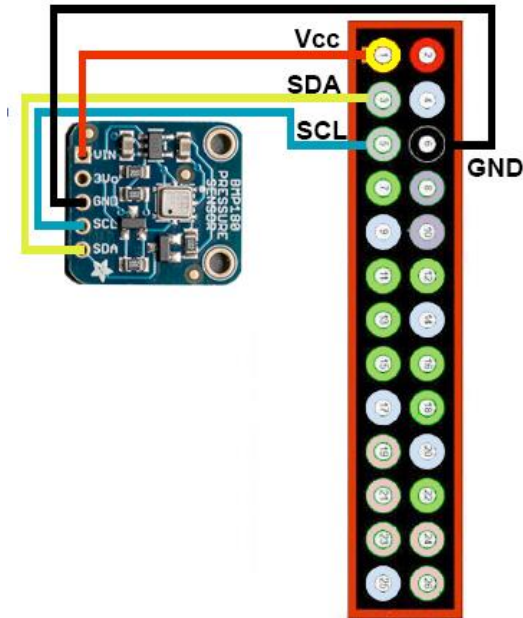
### Montaje en la Raspberry Pi B+.

Dependiendo del tratamiento de la señal que tiene cada uno de los sensores, estos se deben conectar a sus debidos pines de alimentación y transmisión de la información.

La conexión del sensor a los pines GPIO de la Raspberry Pi, se debe realizar de la forma descrita en la Figura 5.

<sup>32</sup> Bosch Invented for life, «BMP180 Digital pressure sensor,» [En línea]. Available: <http://blog.filipeflop.com/embarcados/temperatura-umidade-dht11-com-raspberry-pi.html>. [Último acceso: 19 Abril 2015].

Figura 5. Conexión del sensor de presión atmosférica.



Fuente: CAICEDO Angela. « How to Become an Embedded Developer in Minutes,»<sup>33</sup>

En el momento que se tiene la parte física instalada de forma correcta se procede a realizar los siguientes pasos:

Comentar la línea-i2c bcm2708 del archivo ROSSP-blacklist.conf:

```
pi@raspberrypi:~$ sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

```
#blacklist i2c-bcm2708
```

Añadir i2c-dev en / etc / modules

```
pi@raspberrypi:~$ sudo nano /etc/modules  
i2c-dev
```

Se digitan los siguientes comandos en la consola:

```
sudo modprobe i2c_bcm2708  
sudo modprobe i2c-dev
```

---

<sup>33</sup> CAICEDO Angela, « How to Become an Embedded Developer in Minutes,» [En línea]. Available: <http://www.oracle.com/technetwork/articles/java/ma14-new-to-java-embedded-2177732.html>. [Último acceso: 19 Abril 2015].

Se confirma que los módulos I2C se cargan y están activados, para esto se ejecutan los siguientes comandos:

```
pi@raspberrypi ~ $ lsmod | grep i2c
i2c_dev          6276 0
i2c_bcm2708      4121 0
```

Se deben instalar las siguientes herramientas para el trabajo con I2C de la siguiente manera:

```
pi@raspberrypi:~$ sudo apt-get update
pi@raspberrypi:~$ sudo apt-get install i2c-tools
```

Para la detección del sensor mediante el bus de datos de I2C se debe digitar el siguiente comando:

```
pi@pi-friedrich ~ $ sudo i2cdetect -y 1
```

Al realizar este procedimiento, aparecerá la siguiente línea de código indicando en donde se encuentra el sensor:

```
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- 77
```

Para la lectura de los valores del sensor BMP180 se deben instalar las siguientes librerías que son herramientas fundamentales para la lectura e impresión de los datos:

```
pi@raspberrypi:~$ sudo apt-get install build-essential python-dev python-smbus
pi@raspberrypi:~$ git clone https://github.com/adafruit/Adafruit_Python_BMP.git
pi@raspberrypi:~$ cd Adafruit_Python_BMP
pi@raspberrypi:~$ sudo python setup.py install
```

A continuación se debe crear un documento de texto llamado presión con la siguiente secuencia de comandos:

```
import Adafruit_BMP.BMP085 as BMP085

sensor = BMP085.BMP085()
temp = sensor.read_temperature()
```

```
pressure = sensor.read_pressure()

print 'Temp = {0:0.2f} *C'.format(temp)
print 'Pressure = {0:0.2f} Pa'.format(pressure)
```

Se guardan cambios y se ejecuta la siguiente línea de código:

```
pi @ raspberrypi: ~ $ sudo python presion.py
```

Al realizar la ejecución del archivo creado anteriormente se obtiene la impresión de la medición de la temperatura y humedad visualizada desde la consola de la tarjeta Raspberry Pi.

Brindando la oportunidad de visualizar la interacción del sensor con la interfaz i2C y dando prueba de que la conexión es adecuada con la tarjeta Raspberry Pi.

### **Almacenamiento en la base de datos.**

En primera instancia con el programa descrito anteriormente el sensor solo realizaba la medición de la presión atmosférica y la temperatura a medida que se ejecutaba pero no almacenaba la medición, para poder realizar esta acción, se implementó el siguiente código para almacenar las mediciones generadas por el BMP180:

```
import Adafruit_BMP.BMP085 as BMP085
import MySQLdb
import datetime

sensor = BMP085.BMP085()
temp = sensor.read_temperature()
pressure = sensor.read_pressure()
today = datetime.date.today()

print 'Temperatura = {0:0.2f} *C'.format(temp)
print 'Presion Atmosferica = {0:0.2f} Pa'.format(pressure)

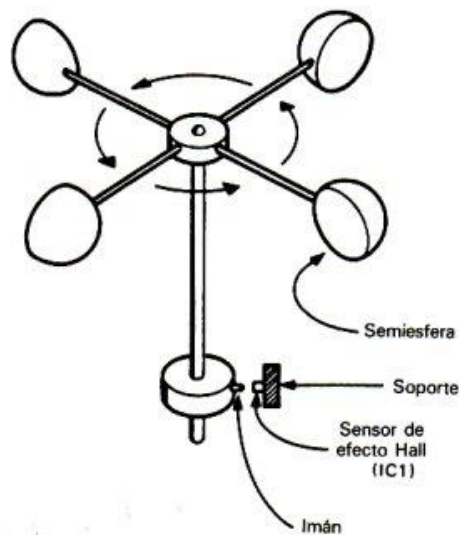
db = MySQLdb.connect("localhost", "root", "tony", "RASPBERRYPI")
curs=db.cursor()

insert = ( " INSERT INTO bmp180 (num_consec, fecha, val_temper, val_pres)"
          " VALUES (%s, %s, %s, %s ) ")
data = ('0',today, temp, pressure )
curs.execute(insert, data)
db.commit()
db.close()
```

### **3.3 ANEMÓMETRO**

Para poder realizar la medición de la velocidad del viento se usa un instrumento llamado anemómetro. La construcción de este instrumento podía hacerse de diferentes formas pero se decidió usar la más adecuada para la integración con la tarjeta Raspberry Pi, pues esta tarjeta a diferencia de Arduino, no posee puertos de lectura análogos, lo cual presenta desventajas a la hora de implementar sensores cuya salida sea de este tipo. Teniendo en cuenta esta limitante se decidió realizar la lectura de esta variable física usando un sensor de efecto hall, el cual funciona como un interruptor que entrega una señal digital que depende de si este sensor detecta o no la presencia de un campo magnético, en nuestro caso este campo lo producirá un imán puesto en una de las hélices del anemómetro, el diseño de este anemómetro se muestra en la figura 6.

Figura 6. Diseño de anemómetro.



Fuente: Mi WEB. Detalles del anemómetro electrónico<sup>34</sup>

El funcionamiento del anemómetro consta de la rotación que hará el imán gracias a la incidencia del viento sobre las semiesferas, al empezar a girar, el sensor de efecto hall entregará cambios de señal cada vez que el imán se cruce enfrente a él. Para poder hacer uso de esta señal de la manera que se desea y lograr medir la velocidad del viento se usarán las revoluciones por minuto, obteniendo la diferencia de tiempo que hay entre una medición y la otra. Luego se hace el procedimiento matemático para obtener la velocidad angular y con esta la velocidad lineal que para nuestro caso es la misma velocidad del viento.

<sup>34</sup> Mi WEB, «Detalles del anemómetro electrónico,» [En línea]. Available: <http://miajas.com/intercom/Anemometro/videoVeleta.htm> [Último acceso: 19 Abril 2015].

Con la siguiente ecuación se obtiene la velocidad angular teniendo en cuenta que "*rpm*" son las revoluciones por minuto del movimiento.

$$\omega = (rpm \times \pi 2)/60$$

Luego de obtener la velocidad angular, se obtiene la velocidad lineal de la siguiente manera, teniendo en cuenta que " $\omega$ " es la velocidad angular y " $r$ " el radio de la hélice del anemómetro.

$$V = \omega r$$

El sensor utilizado para la construcción del anemómetro se llama Modulo De Efecto Hall Con M44 Y Lm393 y se puede observar en la Figura 7.

Figura 7. Módulo de efecto hall.



Fuente: Sensor efecto hall<sup>35</sup>

Este módulo consta de un sensor de efecto hall M44, dispone de una salida digital que puede ser usada como para diferentes aplicaciones como interruptor, contabilizador y en este caso simplemente como señal digital, pero también posee una salida analógica que indica la fuerza que ejerce el campo magnético que también puede ser usada para muchas aplicaciones pero en este caso no será utilizado. Es totalmente compatible con tarjetas de desarrollo Arduino o Raspberry.

El código elaborado en Python para la implementación del anemómetro se puede encontrar en el Anexo B.

La conexión del sensor de efecto hall usado para este anemómetro se puede observar en la figura 8.

---

<sup>35</sup> Sensor efecto hall [en línea] Available: [http://mla-s1-p.mlstatic.com/sensor-magnetico-efecto-hall-arduino-20568-MLA20193704009\\_112014-O.jpg](http://mla-s1-p.mlstatic.com/sensor-magnetico-efecto-hall-arduino-20568-MLA20193704009_112014-O.jpg). [Ultimo acceso:18 de Abril 2015]

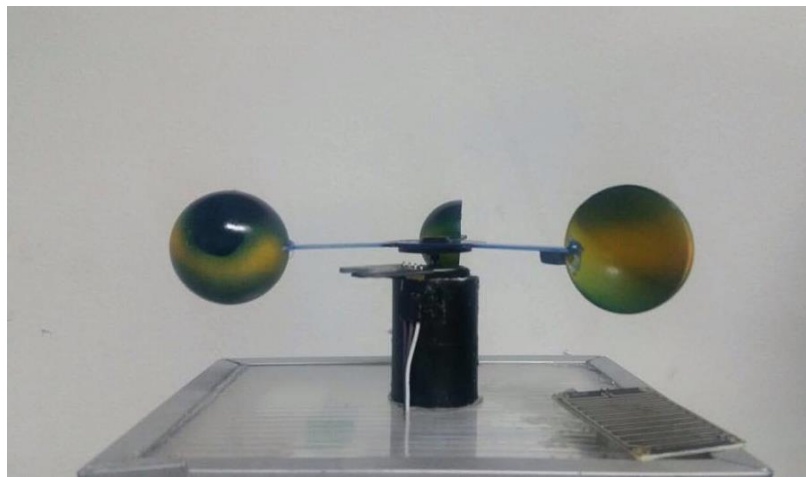
Figura 8. Conexión física del módulo de efecto hall.



Fuente: Del autor.

Se instaló el sensor en la base del anemómetro para lograr la medición de la velocidad del viento como se muestra en la figura 9.

Figura 9. Anemómetro completo.



Fuente: Del autor.

### 3.4 DETECTOR DE LLUVIA

Lograr simplificar los requerimientos de hardware siempre ha sido importante, es por eso que se optó utilizar el módulo YL-83 para la detección de lluvia. Este módulo es capaz de identificar gotas de agua por medio de una serie de pistas conductoras impresas sobre una placa de baquelita (ver Figura 10) cuya separación es bastante pequeña. El módulo básicamente funciona ocasionando

un corto circuito cada vez que estas pistas se mojan. El agua produce un camino resistivo con polaridad positiva entre las pistas que están conectadas a tierra. La corriente que fluye a través del módulo está limitado por resistencias de 10K $\Omega$  en cada conductor, impidiendo que el controlador se vea afectado por el corto circuito.

Figura 10. Placa de baquelita.



Fuente: Panama Hitek Detector de lluvia.<sup>36</sup>

Este módulo también posee un controlador (ver Figura 11) encargado de amplificar el pequeño diferencial de voltaje que se genera cuando el modulo tiene contacto con el agua, generando esa salida que puede ser de tipo análoga o digital que varía entre HIGH y LOW dependiendo de si hay agua o no sobre las pistas antes mencionadas<sup>37</sup> Además tiene un potenciómetro que es usado para su calibración que protege el resto del sistema por el corto que se presenta en el momento que la baquelita tiene contacto con agua. Este controlador logra distinguir la cantidad de resistencia producida y así se lograría determinar la cantidad de agua con la que se tiene contacto, aplicación que funcionaria para medir la pluviometría usando un recipiente en el cual se almacene la cantidad de agua que provenga de la lluvia.

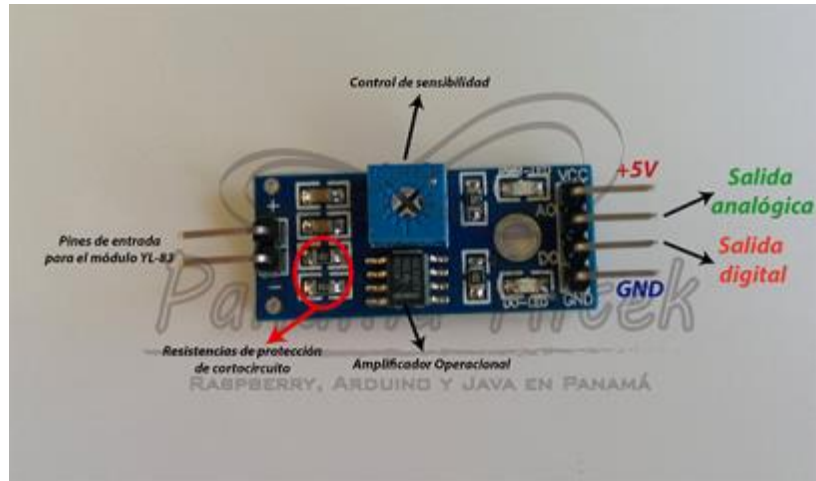
---

<sup>36</sup> Panama Hitek« Detector de lluvia,». [En línea]. Available: <http://panamahitek.com/wp-content/uploads/2014/04/YL-83.png>[Ultimo acceso:18 de Abril 2015]

<sup>37</sup> A. G. González, «Módulo YL-83: Un detector de lluvia,» 6 Abril 2014. [En línea]. Available: <http://panamahitek.com/modulo-yl-83-un-detector-de-lluvia/>. [Último acceso: 25 Abril 2015].



Figura 11. Controlador detector de lluvia.



Fuente: Panama Hitek Detector de lluvia.<sup>38 39</sup>

La salida análoga dependerá de la cantidad de agua que está en contacto.

Montaje Módulo YL-83: Detector de lluvia:

La conexión del sensor en la Raspberry Pi se realizó de la siguiente manera:

1. Sensor VCC (rojo) a 5V Raspberry Pi (+ o férula en un plato)
2. GND del sensor (negro) a la Raspberry Pi GND (o ferrocarril - en el tablero)
3. HACER sensor (verde) para el Raspberry Pi GPIO 21 (pin 40)
4. AO no se conecta en el futuro será el convertidor A / D MCP3008 10-bit 8 canales SPI – DIP

Preparación sensor YL83 con Raspberry Pi:

Para el correcto funcionamiento del sensor en la Raspberry Pi se realizó la previa ejecución de los siguientes comandos, los cuales realizan la instalación de las librerías para el uso del el lenguaje de programación Python, php5 y Mysql. Los comandos usados son los siguientes:

```
sudo aptitude install php5-mysql php5-curl php5-gd php5-idn php-pear php5-imagick php5-imap  
php5-mcrypt php5-memcache php5-mhash php5-ming php5-ps php5-pspell php5-recode php5-  
snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl php5-json
```

<sup>38</sup> Panama Hitek« Detector de lluvia,». [En línea]. Available: <http://panamahitek.com/wp-content/uploads/2014/04/YL-83.png>[Ultimo acceso:18 de Abril 2015]

<sup>39</sup> Panama Hitek «Controlador de detector de lluvia,». [En línea]. Available: <http://panamahitek.com/wp-content/uploads/2014/04/YL-831.png> [Ultimo acceso:18 de Abril 2015].

Adicional la instalación de una librería de interacción de Python con los puertos GPIO se realizó ejecutando el siguiente comando:

```
sudo apt-get install python-rpi.gpio
```

Creación de tabla de almacenamiento de la información:

Para el almacenamiento de la información suministrada por el sensor, fue necesaria la creación de una tabla dentro la base de datos RASPBERRYPI que es en la cual se está almacenando todos los datos recolectados por todos los sensores. El código usado para crear la tabla en la base de datos, que contendrá la información de este sensor es el siguiente:

```
sudo mysql -u root -p
USE RASPBERRYPI;
CREATE TABLE yl83 (
  date DATETIME NOT NULL,
  rain INT NULL);
```

Programa de adquisición de datos del sensor YL83:

El programa es ejecutado en el lenguaje de programación Python y dentro del mismo se realiza la conexión con la base de datos y la tabla creada anteriormente. El código usado para la adquisición se puede visualizar en el Anexo C.

Una vez almacenado el programa, se debe ejecutar y aparecerá en pantalla la fecha y el estado obtenido a través del sensor de lluvia como se puede observar en la figura 12.

Figura 12. Datos obtenidos por el detector de lluvia visualizados por consola.

2015-02-07	15	rain	16
2015-02-07	15	rain	17
2015-02-07	15	rain	18
2015-02-07	15	rain	19
2015-02-07	15	rain	20
2015-02-07	15	rain	21
2015-02-07	15	rain	22
2015-02-07	15	rain	23
2015-02-07	15	rain	24
2015-02-07	15	rain	25
2015-02-07	15	rain	26
2015-02-07	15	rain	27
2015-02-07	15	rain	28
2015-02-07	15	rain	29
2015-02-07	15	rain	30
2015-02-07	15	rain	31
2015-02-07	15	rain	32
2015-02-07	15	rain	33
2015-02-07	15	rain	34
2015-02-07	15	rain	35
2015-02-07	15	rain	36

Fuente: Del autor.

La conexión de los dos dispositivos que conforman el detector de lluvia en la Raspberry Pi se pueden observar en la Figura 13.

Figura 13. Conexión de detector de lluvia en Raspberry Pi.



Fuente: Del autor.

### 3.5 MODULO WIFI

El módulo Wifi se implementó con el objetivo de mejorar la portabilidad del prototipo y también el de brindarle una mejor alternativa de conexión a internet, ya que este prototipo posee integrado un servidor web y una base de datos que pueden ser consultadas desde cualquier lugar del mundo en el caso que así se quisiera, pues tendría toda la plataforma de hardware y software necesarios para lograrlo. En primera medida se debe verificar la versión del Linux para la instalación del driver del módulo WIFI, para hacer esto se digita el siguiente comando:

```
root@raspberrypi:/home/pi# uname -a
```

Version del linux:

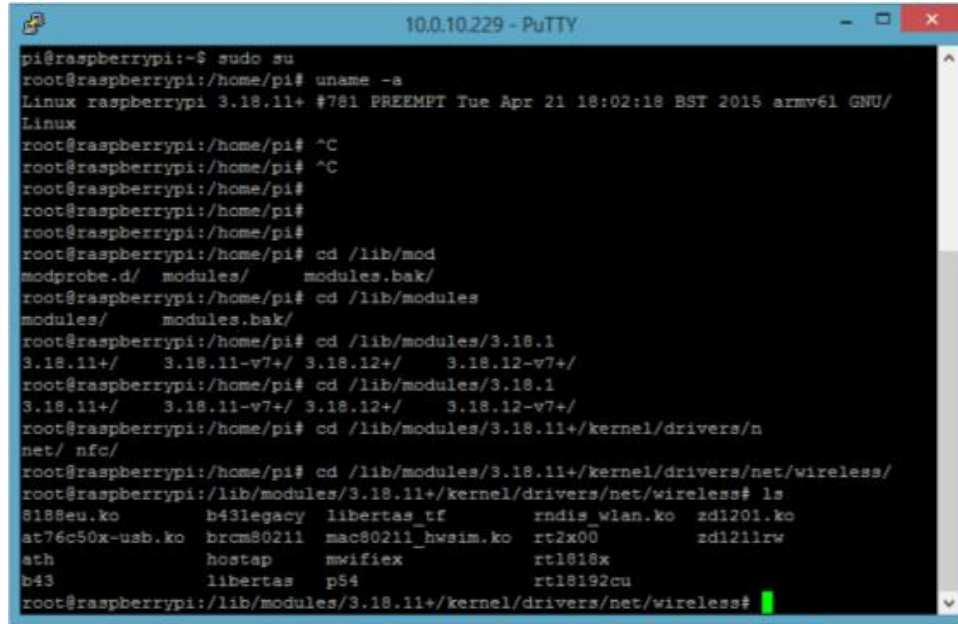
```
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l GNU/
```

Después de verificada la versión del kernel, procedemos a instalar el controlador 8188eu en la siguiente ruta:

```
/lib/modules/3.18.11+/kernel/drivers/net/wireless
```

Este proceso se puede visualizar en la figura 14.

Figura 14. Instalación de controlador 8188eu.



```
10.0.10.229 - PuTTY
pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# uname -a
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015 armv6l GNU/Linux
root@raspberrypi:/home/pi# ^C
root@raspberrypi:/home/pi# ^C
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi#
root@raspberrypi:/home/pi# cd /lib/mod
modprobe.d/ modules/ modules.bak/
root@raspberrypi:/home/pi# cd /lib/modules
modules/ modules.bak/
root@raspberrypi:/home/pi# cd /lib/modules/3.18.1
3.18.11+/ 3.18.11-v7+/ 3.18.12+/ 3.18.12-v7+/
root@raspberrypi:/home/pi# cd /lib/modules/3.18.1
3.18.11+/ 3.18.11-v7+/ 3.18.12+/ 3.18.12-v7+/
root@raspberrypi:/home/pi# cd /lib/modules/3.18.11+/kernel/drivers/n
net/ nfc/
root@raspberrypi:/home/pi# cd /lib/modules/3.18.11+/kernel/drivers/net/wireless/
root@raspberrypi:/lib/modules/3.18.11+/kernel/drivers/net/wireless# ls
8188eu.ko b43legacy libertas_tf rndis_wlan.ko zd1201.ko
at76c50x-usb.ko brcm80211 mac80211_hwsim.ko rt2x00 zd1211rw
ath hostap mwifiex rtl818x
b43 libertas p54 rtl8192cu
root@raspberrypi:/lib/modules/3.18.11+/kernel/drivers/net/wireless#
```

Fuente: Del autor.

Se realiza la instalación del driver dentro de la ruta indicada anteriormente con el siguiente comando:

```
wget https://dl.dropboxusercontent.com/u/80256631/8188eu-0150406.tar.gz
```

Al momento de descargar el driver, el archivo viene comprimido; se debe digitar el siguiente comando para descomprimirlo:

```
tar -xzf 8188eu-0150406.tar.gz
```

Con base en el archivo descomprimido, se extraen 3 archivos que componen el driver, y se debe ejecutar uno de ellos de la siguiente manera:

```
8188eu.conf
8188eu.ko
install.sh
```

```
./install.sh (código de ejecución del archivo)
```

Una vez instalado el driver, es posible visualizar los dispositivos conectados a la Raspberry Pi (ver Figura 15) con el siguiente comando:

```
lsusb
```

Figura 15. Ejecución del comando lsusb

```
root@raspberrypi:/home/pi# cd
root@raspberrypi:~#
root@raspberrypi:~# lsusb
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 0bda:8179 Realtek Semiconductor Corp.
root@raspberrypi:~#
root@raspberrypi:~#
root@raspberrypi:~# depmod -a
```

Fuente: Del autor.

Se verifica que el Modulo WIFI está conectado físicamente, concluyendo la correcta instalación del driver; adicional con el comando “*lsmod*” es posible visualizar los servicios que se están consumiendo en el momento por la Raspberry Pi (ver Figura 16).

Figura 16. Ejecución del comando “*lsmod*”.

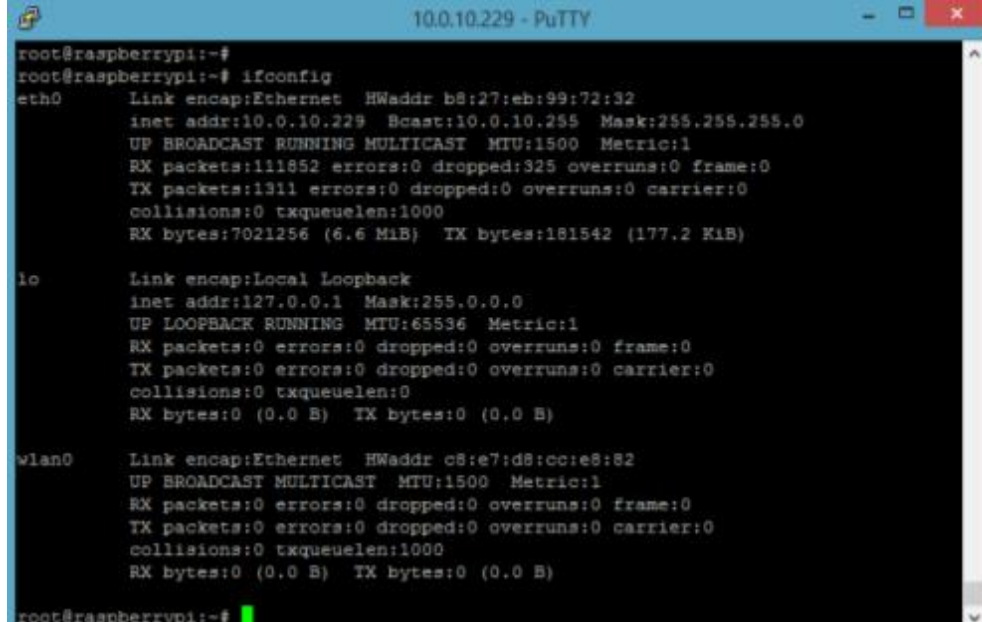
```
root@raspberrypi:~#
root@raspberrypi:~# lsmod
Module                Size  Used by
iptables_filter        1706  0
ip_tables             12393  1 iptables_filter
x_tables              18646  2 ip_tables,iptables_filter
i2c_dev                6709  0
snd_bcm2835            21149  0
snd_pcm               90778  1 snd_bcm2835
snd_seq               61097  0
snd_seq_device         7209  1 snd_seq
snd_timer             23007  2 snd_pcm,snd_seq
snd                   66325  5 snd_bcm2835,snd_timer,snd_pcm,snd_seq,snd_seq_device
i2c_bcm2708            6200  0
spi_bcm2708            6018  0
8188eu                709147  0
uio_pdrv_genirq        3666  0
uio                    9897  1 uio_pdrv_genirq
root@raspberrypi:~#
```

Fuente: Del autor.

El siguiente paso es verificar la interfaz de red con el fin de validar si la tarjeta de red WIFI se encuentra activa, esta validación se hace mediante el siguiente comando (ver Figura 17):

*Ifconfig*

Figura 17. Ejecución del comando “Ifconfig”.



```
10.0.10.229 - PuTTY
root@raspberrypi:~#
root@raspberrypi:~# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:99:72:32
          inet addr:10.0.10.229  Bcast:10.0.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:111852 errors:0 dropped:325 overruns:0 frame:0
          TX packets:1311 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7021256 (6.6 MiB)  TX bytes:181542 (177.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr c8:e7:d8:cc:e8:82
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@raspberrypi:~#
```

Fuente: Del autor.

Al digitar el comando se visualiza la tarjeta de red WIFI denominada **wlan0**, concluyendo que el modulo está listo para su puesta en operación. La conexión de este complemento se puede visualizar en la Figura 18.

Figura 18. Conexión del módulo WIFI en Raspberry Pi.



Fuente: Del autor.

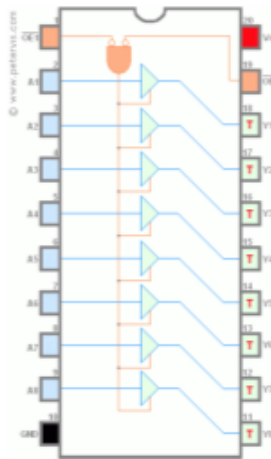


### 3.6 CMOS-TTL

Los pines de la tarjeta Raspberry Pi pueden ser usados como entradas o salidas de señales digitales pero por desgracia los fabricantes no suministran las especificaciones de estos pines adecuadamente y estos son de vital importancia para que tener su la mayor seguridad para el uso de estos pines de manera efectiva. De acuerdo con lo que se describe en la página Mosaic Documentation Web en su documento GPIO Electrical Specifications<sup>40</sup> los pines de la tarjeta Raspberry Pi se asemejan con los del procesador Broadcom BCM2835 que incorpora un núcleo de procesador ARM1176, utilizados en teléfonos inteligentes, televisores, y otros dispositivos. Los pines GPIO de la tarjeta Raspberry Pi son muy versátiles y se puede configurar mediante software el control de la transmisión de corriente que va de 2mA a 16mA en incrementos de 2mA, pero por desgracia, no se puede limitar la corriente máxima y su mal uso puede dañar los pines.

Con el objetivo de proteger los pines de la tarjeta, se tuvo en cuenta que las tensión máxima no puede superar los 3.3 V, así que las señales debían representar un 1 lógico en esta tensión y un 0 lógico en 0 V haciendo uso de un circuito CMOS a TTL por medio del chip búfer octal 74AHCT541 que se puede ver en la figura 19.

Figura 19. Chip 74AHCT541



Fuente: Raspberry Pi GPIO: CMOS to TTL logic converter.<sup>41</sup>

<sup>40</sup> Mosaic Documentation Web, «GPIO Electrical Specifications,» [En línea]. Available: <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>. [Último acceso: 25 07 2015].

<sup>41</sup> J. Peter « Raspberry Pi GPIO: CMOS to TTL logic converter,» [En línea]. Available: [http://www.petervis.com/Raspberry\\_Pi/CMOS\\_to\\_TTL\\_Logical\\_Level\\_Converter/CMOS\\_to\\_TTL\\_Logical\\_Level\\_Converter.html](http://www.petervis.com/Raspberry_Pi/CMOS_to_TTL_Logical_Level_Converter/CMOS_to_TTL_Logical_Level_Converter.html) [Último acceso: 25 07 2015].

Estos chips AHCT pueden resolver estos tipos de problemas de interfaz, es usado con voltajes bajos como 2V pero usualmente se usa para voltajes entre los 3.3 y 5 Voltios permitiendo ser conectado directamente a cualquier pin de la Raspberry Pi.<sup>42</sup>

### 3.7 PROTOTIPO

La integración de los diferentes sensores en la tarjeta Raspberry Pi dieron lugar al prototipo de estación meteorológica planteada, pero esta necesitaba un recipiente en la que se pudiera proteger, por esto se construyó el contenedor del sistema con policarbonato el cual es resistente a la deformación térmica, es un gran aislante eléctrico, tiene una gran resistencia a impactos, es bastante ligero y es usado para diferentes aplicaciones industriales. Su resistencia al clima lo hace ser utilizado en cubiertas, ya que puede resistir temperaturas extremas de -40 a 100 grados centígrados. La estación meteorológica se puede ver en la figura 20.

Figura 20. Diseño final del Prototipo.



Fuente: Del autor.

---

<sup>42</sup> P. J. Vis, «Raspberry Pi GPIO: CMOS to TTL Logic Level Converter Circuit,» [En línea]. Available: [http://www.petervis.com/Raspberry\\_PI/CMOS\\_to\\_TTL\\_Logic\\_Level\\_Converter/CMOS\\_to\\_TTL\\_Logic\\_Level\\_Converter.html](http://www.petervis.com/Raspberry_PI/CMOS_to_TTL_Logic_Level_Converter/CMOS_to_TTL_Logic_Level_Converter.html). [Último acceso: 28 Julio 2015].



#### **4. ANÁLISIS COMPARATIVO DE LOS COSTOS ENTRE LA SOLUCIÓN PLANTEADA Y LAS EXISTENTES EN EL MERCADO**

La importancia de los costos a la hora de desarrollar un sistema basado en las tecnologías de la información, es básicamente lo que define la viabilidad del mismo, es por eso que se debe tener en cuenta cuales son las tecnologías existentes, sus costos, su capacidad, su precisión en las medidas y otras más características que puedan presentar la oportunidad de hacer que un prototipo sea más competitivo e innovador.

El costo de una estación meteorológica está definido por la cantidad de variables físicas que se desean medir, pues esto implica que se deban implementar más sensores, algunos más costosos que otros. Además, la precisión de las mediciones también se convierte un factor muy importante porque el precio de un sensor varía dependiendo su sensibilidad y precisión.

Teniendo en cuenta esos factores mencionados, se comparan las capacidades y costos que poseen las estaciones que se pueden encontrar actualmente en el mercado, contra las del prototipo diseñado.

##### **4.1 ESTACIONES EXISTENTES.**

###### **4.1.1 Estación meteorológica WatchDog WD 2700**

Esta opción de estación meteorológica que se encuentra en el mercado, tiene la capacidad de medir evo transpiración, radiación solar, dirección del viento, velocidad del viento, temperatura, humedad, y pluviosidad. Posee funciones mucho más complejas que anterior como medir 5 parámetros de forma simultánea, puede expandirse a 6 sensores externos más, su memoria interna permite guardar intervalos de tiempo de 30min por 180 días, es estable aun con batería baja, se puede programar directamente o con un software opcional, y tiene la posibilidad de uso móvil. Su costo es de alrededor 1.566,95 €. La estación meteorológica se puede visualizar en la Figura 21.

Figura 21. Estación meteorológica WatchDog WD 2700



Fuente: Estacion meteorológica WD-2900. PCE Instruments<sup>43</sup>

#### 4.1.2 Estación meteorológica WD-2900.

Esta estación meteorológica (ver Figura 22) ofrece las mismas funciones que la anterior con un plus con respecto a su alimentación. Ofrece una operatividad de hasta 12 meses con alimentación por medio de batería, puede ampliar solamente 5 sensores más y su costo de aproximadamente 2.171,95 €.

Figura 22. Estación meteorológica WD-2900



---

<sup>43</sup> «Estación meteorológica WD-2900,». [En línea]. Available: [https://www.pce-instruments.com/espanol/instrumento-medida/medidor/estacion-meteorologica-spectrum-technologies-inc.-estaci\\_n-meteorol\\_gica-watchdog-wd-2700-det\\_96964.htm?\\_list=k&\\_listpos=2](https://www.pce-instruments.com/espanol/instrumento-medida/medidor/estacion-meteorologica-spectrum-technologies-inc.-estaci_n-meteorol_gica-watchdog-wd-2700-det_96964.htm?_list=k&_listpos=2). [Ultimo acceso: 18 de Abril 2015].

Fuente: Estacion meteorológica wmr300. PCE Instruments<sup>44</sup>

#### 4.1.3 Estación meteorológica WMR300 de Oregon Scientific.

La estación WMR300 (ver figura 23), es una estación profesional de ultra precisión que cuenta con mediciones de temperatura, humedad, velocidad del viento, dirección del viento, presión atmosférica y lluvia. Los intervalos de medición personalizados pueden capturar y almacenar datos. Cuenta con un display en el cual se pueden visualizar datos obtenidos desde hace 24 meses y cuenta con un transmisor solar el cual tiene un alcance de 300 metros. Su costo es de aproximadamente 699,90 €.<sup>45</sup>

Figura 23. Estación WMR300



Fuente: Estación Meteorológica Ultra-Precisión. Oregon Scientific.<sup>46</sup>

<sup>44</sup> «Estación meteorológica wmr300,». [En línea]. Available: [https://www.pce-instruments.com/espanol/instrumento-medida/medidor/estacion-meteorologica-spectrum-technologies-inc.-estaci\\_n-meteorol\\_gica-wd-2900-det\\_388701.htm?\\_list=kat&\\_listpos=3](https://www.pce-instruments.com/espanol/instrumento-medida/medidor/estacion-meteorologica-spectrum-technologies-inc.-estaci_n-meteorol_gica-wd-2900-det_388701.htm?_list=kat&_listpos=3). [Último acceso: 18 de Abril 2015].

<sup>45</sup> «Estación meteorológica Davis Vantage Pro 2 Aktiv Plus 6163EU,». [En línea]. Available: <http://global.oregonscientific.com/manual/WMR300.pdf>. [Último acceso: 18 de Abril 2015].

<sup>46</sup> Oregon Scientific, «Ultra Professional Weather System,» 22 Marzo 2006. [En línea]. Available: <http://global.oregonscientific.com/manual/WMR300.pdf>. [Último acceso: 10 Mayo 2015].

#### 4.1.4 Estación meteo Davis Vantage Pro 2 Aktiv Plus 6163EU.

Esta estación meteorológica es la más completa de gama Vantage Pro 2, es ideal para la industria, agricultura y ganadería. Posee un pluviómetro extra grande para la precipitación y cuenta con un anemómetro ultra preciso que puede medir la velocidad y dirección del viento. Su sensor de humedad y temperatura está perfectamente ventilado para obtener fidelidad en la medición. Su alimentación está centrada en una placa solar con batería y sus datos son transmitidos por radio en la frecuencia de 868Mhz<sup>47</sup>, usando el protocolo FHSS de salto de frecuencia que permite una distancia máxima de 300 metros. Su precio aproximado es de 1.239,00 €.<sup>48</sup>

Esta estación posee una base receptora (ver Figura 24) y una estación base (ver Figura 25).

Figura 24. Base receptora.



Fuente: Davis Instruments, «Integrated Sensor Suite with,»

<sup>47</sup> Davis Instruments, «Integrated Sensor Suite with,» 22 Marzo 2006. [En línea]. Available: [http://www.davisnet.com/product\\_documents/weather/manuals/VP2\\_Fan\\_Aspirated\\_ISS\\_Rev\\_C.pdf](http://www.davisnet.com/product_documents/weather/manuals/VP2_Fan_Aspirated_ISS_Rev_C.pdf). [Último acceso: 10 Mayo 2015].

<sup>48</sup> TFV-Solar, «Ahorro energético-Vida sostenible-Nuevas energías,» [En línea]. Available: [http://www.tiendafotovoltaica.es/epages/61359426.sf/es\\_ES/?ObjectPath=/Shops/61359426/Products/%22Davis%20Vantage%20Pro%202%20Aktiv%20Plus%206163EU%22](http://www.tiendafotovoltaica.es/epages/61359426.sf/es_ES/?ObjectPath=/Shops/61359426/Products/%22Davis%20Vantage%20Pro%202%20Aktiv%20Plus%206163EU%22). [Último acceso: 15 Junio 2015].

Figura 25. Estación base.



Fuente: Davis Instruments, «Integrated Sensor Suite with,»

#### 4.2 COMPARACIÓN DE COSTOS.

Es importante tener en cuenta cuales son las tecnologías existentes para conocer los aspectos a mejorar. El prototipo planteado se desarrollo teniendo en cuenta las capacidades que poseen las estaciones anteriormente mencionadas, pero su objetivo principal, es el de mejorar estas prestaciones y disminuir considerablemente el costo, en la tabla 5 se pueden observar la diferencia en costos entre las estaciones que actualmente se encuentran en el mercado y el prototipo diseñado producido en serie. Se hizo aproximaciones de los precios en pesos colombianos sin tener en cuenta los gastos de importacion e impuestos relacionados.

Tabla 5. Comparación de costos entre tecnologías existentes y prototipo.

Estación Meteorológica	WatchDog 2700	WD WD-2900	WMR-300 OREGON SCIENTIFIC	Davis Pro 2	Vantage Prototipo producido en serie.	UPC
Variables físicas que puede medir.	Temperatura Humedad Dirección del viento Velocidad del viento Presión atmosférica Pluviometría Radiación solar Evo transpiración	Temperatura Humedad Dirección del viento Velocidad del viento Presión atmosférica Pluviometría Radiación solar Evo transpiración	Temperatura Humedad Dirección del viento Velocidad del viento Presión atmosférica Pluviometría -	Temperatura Humedad Dirección del viento Velocidad del viento Presión atmosférica Pluviometría Radiación solar Evo transpiración	Temperatura Humedad - Velocidad del viento Detección de lluvia Presión Atmosférica -	
Costo	\$3.800.000.	\$6.320.000.	\$2.011.694	\$3.600.000	\$500.000	

Fuente: Del autor.

Cada uno de los sensores cuesta alrededor de \$20.000 pesos y se pueden encontrar fácilmente en el mercado, por lo tanto la implementación de más sensores no afecta mucho el costo del desarrollo del prototipo y la tarjeta Raspberry Pi cuenta con la capacidad de conectarse a mas sensores ya que se usaron 13 pines de sus 40 en total. El desarrollo del prototipo puede aumentar el costo si se considera la investigación necesaria para su implementación, al punto de que este llegue a ser comparable con el de las tecnologías existentes, pero como se muestra en la tabla anterior se habla del precio de la estación en el caso de que su fabricación se realice en serie.

Aunque las otras estaciones puedan más variables físicas, no todas poseen la integración de un servidor web, ni tampoco la implementación de una base de datos que les permita presentar las mediciones a través de internet como el prototipo desarrollado si lo realiza.

## 5. PRUEBAS DE VALIDACIÓN DEL PROTOTIPO

En este capítulo se describen las pruebas de cada uno de los sensores implementados, las mediciones obtenidas por cada uno de ellos y la visualización de las mediciones desde la aplicación web, con el objetivo de mejorar la eficiencia operativa del prototipo y validar su funcionamiento.

### 5.1 HORA DEL SERVIDOR.

**Realizada por:** Anthony Rojas Paternina.

**Modulo:** Todos.

**Lugar:** Universidad Piloto de Colombia.

**Fecha:** 19/08/2015.

#### **Descripción de la prueba:**

Verificar mediante líneas de código si la hora del servidor coincide con la hora actual.

#### **Resultados esperados:**

Se espera que la hora que tiene almacenada el servidor sea idéntica a la hora actual del lugar donde se encuentra la estación meteorológica móvil.

#### **Resultados obtenidos:**

Se verificó que en el momento de apagar la estación meteorológica la hora actual se pierde; una vez se enciende de nuevo la Raspberry Pi la hora queda predeterminada y es necesario volver a ajustar la hora para que en el momento de que los sensores realicen la medición los datos sean correctos

### 5.2 PRUEBA DE CONEXIÓN WIFI.

**Realizada por:** Anthony Rojas Paternina.

**Modulo:** Wifi.

**Lugar:** Universidad Piloto de Colombia.

**Fecha:** 19/08/2015.

#### **Descripción de la prueba:**

Verificar conexión de la Raspberry Pi a una red inalámbrica colocando en funcionamiento el Módulo WIFI.

**Resultados esperados:**

Se espera que la Raspberry Pi pueda tener acceso a Internet mediante una conexión inalámbrica.

**Resultados obtenidos:**

Se verificó una correcta conexión de la Raspberry Pi a una red LAN; para la conexión fue necesario colocar el Nombre de la Red y la Contraseña mediante consola; no fue posible realizar la conexión mediante la interfaz gráfica.

**5.3 PRUEBA PRESIÓN ATMOSFÉRICA**

**Realizada por:** Anthony Rojas Paternina.

**Modulo:** Sensor Presión Atmosférica

**Lugar:** Calle 63 Cra.13 Grupo OET

**Fecha:** 19/08/2015.

**Descripción de la prueba:**

Verificar medición correcta de la presión atmosférica realizando mediciones en Facatativá con una altura baja y Bogotá en el edificio de Seguros Bolívar piso 11.

**Resultados esperados:**

Se espera que la Raspberry Pi pueda tener acceso a Internet mediante una conexión inalámbrica.

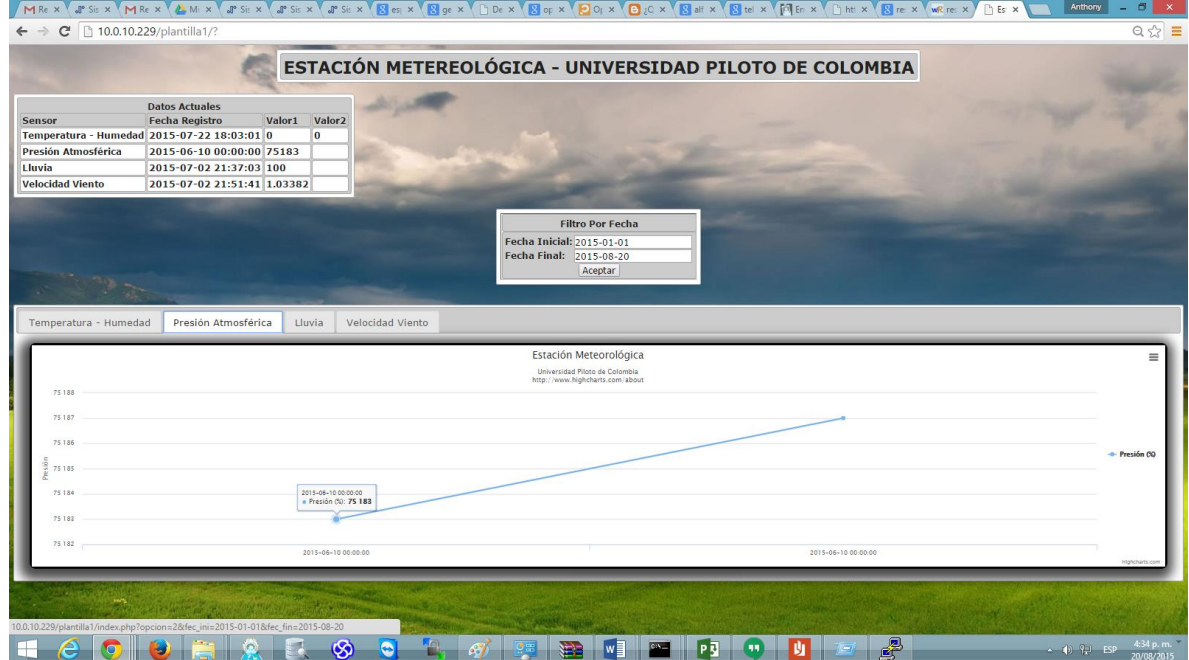
**Resultados obtenidos:**

El sensor realizó una medición errónea de la información, siempre mostraba la misma medición; se verificó la implementación del programa y estaba correcto; el siguiente paso fue validar la conexión del sensor con la Raspberry Pi y el PIN GPIO que transmite la información estaba conectado de forma errónea. Se realizó el cambio del pin y se modificó la información en el programa. Una vez ejecutado el procedimiento anterior se realizaron las pruebas de nuevo verificando una medición correcta.

**Imagen de la prueba:**



Figura 26. Prueba sensor BMP180.



Fuente: Propia del autor.

## 5.4 PRUEBA CAMBIOS DE TEMPERATURA

**Realizada por:** Anthony Rojas Paternina y Daniel Murcia Almanza.

**Modulo:** Sensor Dht11.

**Lugar:** Universidad Piloto de Colombia.

**Fecha:** 15/05/2015.

### Descripción de la prueba:

Verificar cambio en la medición del sensor en lugares con temperaturas diferentes.

### Resultados esperados:

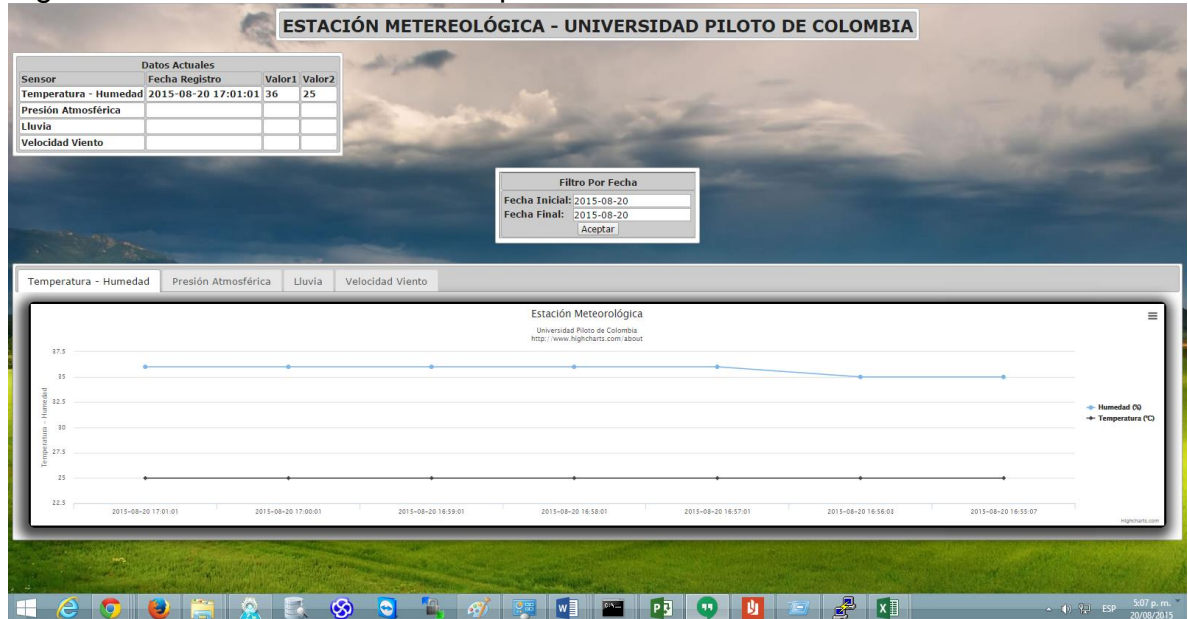
Se espera que la Raspberry Pi muestre en pantalla los datos del cambio de temperatura al realizar la medición en distintos lugares.

### Resultados obtenidos:

El sensor se colocó en la temperatura ambiente en Facatativá, Cundinamarca, allí marco 20 grados Centígrados dentro de la casa; luego se realizó el mismo procedimiento abriendo la nevera y colocando el sensor dentro de la misma, realizando una medición de 3 grados centígrados.

Imagen de la prueba:

Figura 27. Prueba cambios de temperatura.



Fuente: Propia del autor.

## 5.5 PRUEBA CAMBIOS DE HUMEDAD.

Realizada por: Anthony Rojas Paternina y Daniel Murcia Almanza.

Modulo: Sensor Presión Atmosférica

Lugar: Universidad Piloto de Colombia.

Fecha: 15/05/2015.

### Descripción de la prueba:

Verificar cambio en la medición del sensor en condiciones aleatorias.

### Resultados esperados:

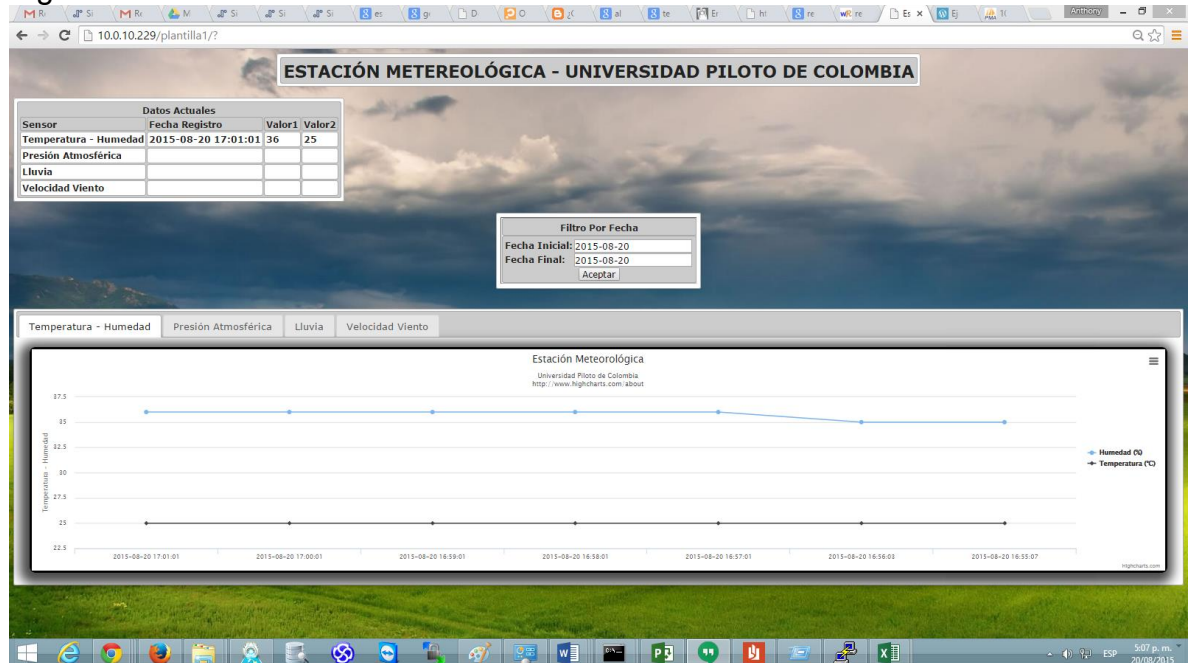
Se espera que la Raspberry Pi muestre en pantalla los datos del de humedad en el momento que se coloque un trapo húmedo en el sensor.

### Resultados obtenidos:

Se colocó un trapo húmedo en el sensor de humedad, y se verifico en pantalla el cambio elevado del porcentaje de humedad, verificando el correcto funcionamiento del Dth11.

**Imagen de la prueba:**

**Figura 29. Prueba de medición de humedad.**



Fuente: Propia del autor.

## 5.6 PRUEBA DETECCIÓN DE LLUVIA

**Realizada por:** Anthony Rojas Paternina y Daniel Murcia Almanza.

**Modulo:** YL-83.

**Lugar:** Universidad Piloto de Colombia.

**Fecha:** 19/07/2015.

### Descripción de la prueba:

Verificar cambio en la medición del sensor realizando la prueba en dos condiciones: lluvia y no lluvia.

### Resultados esperados:

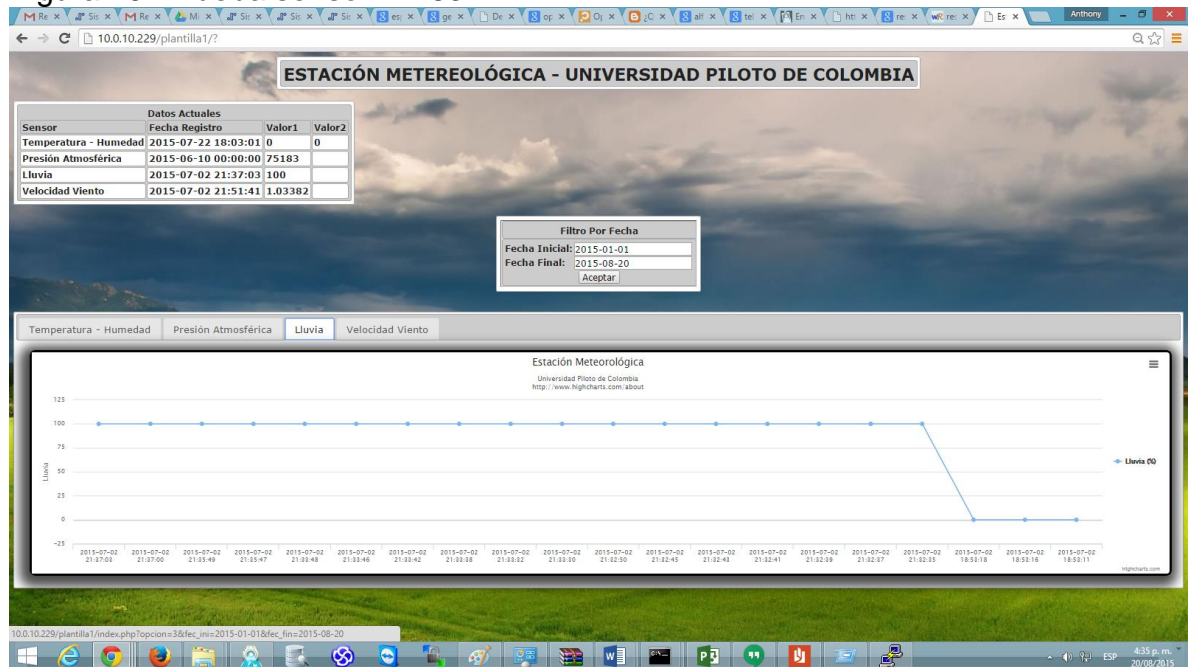
Se espera que la Raspberry Pi muestre en pantalla la medición del sensor de lluvia a partir de los factores que se le apliquen: llueve o no llueve.

### Resultados obtenidos:

Al momento de dejar caer gotas sobre el sensor en pantalla indicaba que no llovía, siendo un resultado falso; se verificó el código y la compilación se realizó de forma correcta; al validar la conexión física se percibió que el pin GPIO indicado en el código para la ejecución del programa no coincidía con el conectado. Se corrigió el error, se implementó de nuevo la prueba con una respuesta verdadera y satisfactoria.

**Imagen de la prueba:**

**Figura 29. Prueba sensor YL-83.**



Fuente: Propia del autor.

## 5.7 PRUEBA VELOCIDAD DEL VIENTO

**Realizada por:** Anthony Rojas Paternina y Daniel Murcia Almanza.

**Modulo:** Sensor Presión Atmosférica

**Lugar:** Universidad Piloto de Colombia.

**Fecha:** 19/07/2015.

### Descripción de la prueba:

Verificar cambio de resultado de la velocidad a medida que el imán toca más o menos veces el sensor.

## Resultados esperados:

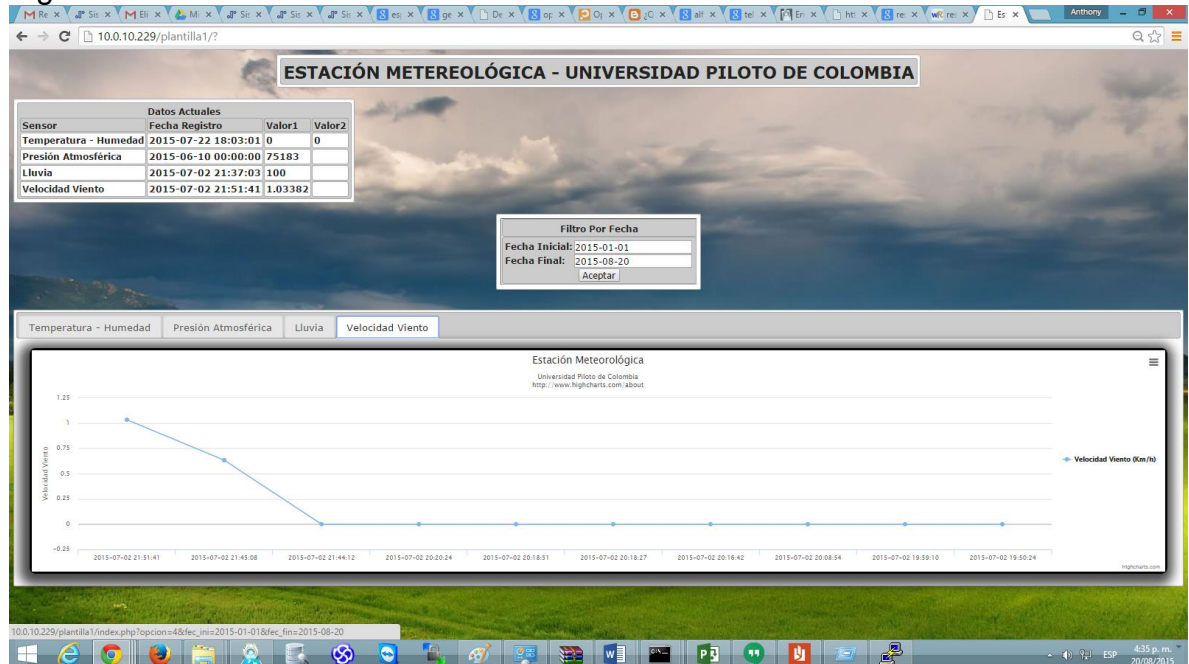
Se espera que la Raspberry Pi muestre en pantalla una velocidad mayor si el imán toca muchas veces el sensor, mientras que se espera una menor velocidad si el imán toca menos veces el sensor.

## Resultados obtenidos:

Al momento de realizar la prueba, el resultado de la velocidad siempre era 0; se validó el código y se halló el error en el método que realizaba el cálculo de la velocidad, se corrigió el error, se volvió a realizar la prueba y funciono correctamente.

## Imagen de la prueba:

Figura 30. Prueba medición del viento.



Fuente: Propia del autor.

## CONCLUSIONES

Las estaciones meteorológicas existentes tienen un costo elevado debido a la precisión y cantidad de variables físicas que miden, pero es importante resaltar que no todas poseen la capacidad de integrar sus mediciones con una base de datos que permita el acceso remoto a su información recolectada a través de una aplicación web de manera que si lo hace el prototipo desarrollado en este proyecto, teniendo en cuenta que la implementación de más sensores en el prototipo no requiere de un aumento considerable en sus costos.

El costo de desarrollo del prototipo puede ser comparable con las demás estaciones existentes, pero la fabricación en serie de este puede disminuir el costo considerablemente, lo que haría que este modelo de estación meteorológica tuviera un costo realmente bajo considerando los servicios que presta, la personalización que permite, que la adaptación de mayor cantidad de sensores es totalmente posible y que también, con respecto a la investigación necesaria para su desarrollo, se requirió de tiempo y conocimiento en diferentes lenguajes de programación como C y Python, lenguajes usados para la elaboración del código fuente que interpreta la información adquirida por cada uno de los sensores, y cuyos conocimientos de estos lenguajes de programación se adquirieron solamente durante el desarrollo de este proyecto.

El conocimiento en Linux para el desarrollo del prototipo fue crucial, ya que sobre este sistema operativo se implementaron los complementos necesarios para lograr la adecuada conexión e interacción de cada sensor con la tarjeta Raspberry Pi. En el ámbito de las telecomunicaciones tener conocimientos sobre este sistema operativo y los comandos que usa, es de vital importancia, debido a que la mayoría de los servidores lo usan. Por lo tanto, es importante que los estudiantes posean un acercamiento y conocimiento en estas áreas para dar lugar a desarrollos como este y mejorar su perfil profesional.

El prototipo desarrollado demostró que tiene la capacidad suficiente para lograr medir más de 5 variables físicas, como también puede soportar un servidor web que permite el acceso remoto a la información recolectada y además, posee la capacidad necesaria para ampliar su alcance al punto de realizar predicción del clima, brindando la posibilidad de que el proyecto sea retomado por parte del semillero de investigación.

## REFERENCIAS

- [1] Programa de las Naciones Unidas para el Desarrollo en Colombia- PNUD, «El cambio climático en Colombia y en el sistema de las Naciones unidas,» 5 Enero 2010. [En línea]. Available: [http://www.pnud.org.co/img\\_upload/61626461626434343535373737353535/Brochure%20resumen%20Proyecto.pdf](http://www.pnud.org.co/img_upload/61626461626434343535373737353535/Brochure%20resumen%20Proyecto.pdf). [Último acceso: 16 02 2015].
- [2] M. Cote, P. Martin, J. Gonzales I. y A. Cardona, «El cambio climatico en Colombia y en el sistema de las Naciones Unidas,» PNUD, New York, 2010.
- [3] R. M. Rodriguez, A. Capa, A. Portela, F. Martín, J. Quirantes y F. Rey, «¿Que es la meteorología?,» de *METEREOLOGÍA Y CLIMATOLOGIA*, España, Fundación Española para la ciencia y la tecnología, 2004, pp. 612-620.
- [4] Organizacion de Estados Iberoamericanos - OEI, «el cambio climático: ¿Que es?,» 10 Febrero 2006. [En línea]. Available: [http://www.oei.es/decada/portadas/climate\\_change\\_youth\\_es.pdf](http://www.oei.es/decada/portadas/climate_change_youth_es.pdf). [Último acceso: 6 Marzo 2015].
- [5] D. F. Valencia Z., *Diseño y contrucción de un sistema de adquisición de la temperatura, humedad, velocidad y direccion del viento para la evaluacion del potencial micro-elico*, Cali: Universidad de San Buenaventura, 2013.
- [6] K. C. Laudon y J. P. Laudon, *Sistemas de información Gerencial*, Reino Unido: Pearson Educacion, 2001.
- [7] Wikipedia, «Wikipedia la enciclopedia libre,» Fundación Wikimedia, Inc., Enero 1990. [En línea]. Available: <http://es.wikipedia.org/wiki/Temperatura>. [Último acceso: 2 Febrero 2015].
- [8] Pontificia Universidad Católica de Chile, «Clasificación termica,» 20 Diciembre 1961. [En línea]. Available: [http://www7.uc.cl/sw\\_educ/hort0498/HTML/p015.html](http://www7.uc.cl/sw_educ/hort0498/HTML/p015.html). [Último acceso: 4 Marzo 2015].
- [9] ECOLOGIAHOY, «Humedad absoluta, especifica y relativa,» Ecologiahoy.com, 12 Julio 2010. [En línea]. Available: <http://www.ecologiahoy.com/humedad-absoluta-especifica-y-relativa>. [Último acceso: 22 Enero 2015].
- [10] INTERNATIONAL, «¿Que es la humedad relativa?,» 3 Agosto 2011. [En línea]. Available: [https://www.akzonobel.com/terms\\_of\\_use.aspx](https://www.akzonobel.com/terms_of_use.aspx). [Último acceso: 3 Enero 2015].
- [11] Asistencia Técnica Agrícola , «INFORMACIÓN GENERADA PARA LOS AGRICULTORES POR AGRO TECNOLOGIA TROPICAL,» FerozoSite, Septiembre 2013. [En línea]. Available: [http://www.agro-tecnologia-tropical.com/la\\_humedad.html](http://www.agro-tecnologia-tropical.com/la_humedad.html). [Último acceso: 24 Abril 2015].
- [12] EDUCLAB, «Dinamica atmosférica,» Gobierno de España, 30 enero 2000. [En línea]. Available:



- [http://recursostic.educacion.es/secundaria/edad/1esobiologia/1quincena5/1q5\\_contenidos\\_3f.htm](http://recursostic.educacion.es/secundaria/edad/1esobiologia/1quincena5/1q5_contenidos_3f.htm). [Último acceso: 2 Abril 2015].
- [13] fedora project website, «capítulo 7. el Viento,» 23 Enero 2009. [En línea]. Available: <http://200.58.146.28/nimbus/weather/pdf/cap7.pdf>. [Último acceso: 23 Abril 2015].
  - [14] EDULAB - Ministerio de Educación de España, «El viento,» 23 Enero 2010. [En línea]. Available: <http://educalab.es/intef>. [Último acceso: 25 Abril 2015].
  - [15] Universidad Politecnica de Madrid, «Viento y sus efectos sobre los cultivos,» powered by eduCommons, 2 Octubre 2013. [En línea]. Available: <http://ocw.upm.es/>. [Último acceso: 15 Abril 2015].
  - [16] Wikipedia la enciclopedia libre, «Tornado,» Fundación Wikimedia, Inc., 2 diciembre 1980. [En línea]. Available: <http://es.wikipedia.org/wiki/Tornado>. [Último acceso: 3 abril 2015].
  - [17] Comunidad arduino en español- ARDUINOSTAR.COM, «Sensor de temperatura y humedad DHT 11,» Comunidad arduino en español, 2012. [En línea]. Available: <http://rduinostar.com/>. [Último acceso: 2 Abril 2015].
  - [18] D. Fraj B., «Blog personal David Fraj Blesa,» davidfracj@gmail.com, 28 Abril 2013. [En línea]. Available: <http://www.davidfracj.com/?m=0>. [Último acceso: 22 Abril 2015].
  - [19] A. Garcia y K. Navarro, «DHT 11. sensor de humedad y temperatura para Arduino,» Panamahitek, 14 Febrero 2013. [En línea]. Available: <http://panamahitek.com/dht11-sensor-de-humedadtemperatura-para-arduino/>. [Último acceso: 28 Abril 2015].
  - [20] SEEED WIKI, «Grove - Temperature and Humidity Sensor,» SEEED WIK, 12 Marzo 2003. [En línea]. Available: [http://www.seeedstudio.com/wiki/Grove-Temperature\\_and\\_Humidity\\_Sensor](http://www.seeedstudio.com/wiki/Grove-Temperature_and_Humidity_Sensor). [Último acceso: 23 Febrero 2015].
  - [21] RASPBERRY PI FONDATION , «RASPBERRY PI FONDATION,» About us of raspberry pi, 23 Abril 2010. [En línea]. Available: <http://www.raspberrypi.org>. [Último acceso: 2015 Febrero 18].
  - [22] CARLETTO. J. et al. , «Datos Climaticos On line con software libre,» Universidad Nacional de San Luis, Argentina, 2007.
  - [23] D. e. a. SCHVARTZMAN, «Sistema de comunicación y transmisión de datos desde estaciones metereologicas,» de *Sistema de comunicación y transmisión de datos desde estaciones metereologicas*, Brasil, Iberoamerican Journal of Industrial Engineering Vol 4, 2012.
  - [24] B. MONTALVO, «Prototipo didáctico de una estación meteorológica monitoreada a distancia,» Instituto Politécnico Nacional, México, 2013.
  - [25] S. e. a. VELÁSQUEZ, «Monitoreo de variables analógicas usando Raspberry pi,» Universidad, ciencia y tecnología, Diciembre 2013. [En línea]. Available: <http://www.poz.unexpo.edu.ve/postgrado/uct/revista/index.php/uct/article/viewFile/267/218>. [Último acceso: 18 Febrero 2015].



- [26] F. G. VÁSCONEZ ALBÁN, «Diseño e implementación de un prototipo de estación de medición de parámetros meteorológicos para su incorporación en una red extendida de monitorización ambiental,» Escuela Politecnica Nacional, Quito, 2009.
- [27] Universidad Autónoma Chipango, "Precipitación," [Online]. Available: [http://www.chapingo.mx/irrigacion/planest/documentos/apuntes/hidrologia\\_susp/UNIDAD4-PRECIPITACION.pdf](http://www.chapingo.mx/irrigacion/planest/documentos/apuntes/hidrologia_susp/UNIDAD4-PRECIPITACION.pdf). [Accessed 19 Abril 2015].
- [28] Georgia State University, "Humedad Relativa," HyperPhysics, [Online]. Available: <http://hyperphysics.phy-astr.gsu.edu/hbasees/kinetic/relhum.html>. [Accessed 20 Marzo 2015].
- [29] International, "¿Qué es la "Humedad Relativa"?", [Online]. Available: <http://www.yachtpaint.com/esp/diy/ask-the-experts/qu%C3%A9-es-la-humedad-relativa.aspx>. [Accessed 20 Marzo 2015].
- [30] Definicion.De, "Definicion de Presión Atmosferica," [Online]. Available: <http://definicion.de/presion-atmosferica/>. [Accessed 1 abril 2015].
- [31] M. C. Carpio, "Precipitación," [Online]. Available: <http://www.olivacordobesa.es/Precipitaci%F3n.pdf>. [Accessed 15 Abril 2015].
- [32] A. G. González, "Módulo YL-83: Un detector de lluvia," 6 Abril 2014. [Online]. Available: <http://panamahitek.com/modulo-yl-83-un-detector-de-lluvia/>. [Accessed 25 Abril 2015].
- [33] "RaspberryPi.org," 3 Febrero 2012. [Online]. Available: <https://www.raspberrypi.org/forums/viewtopic.php?p=462982#p462982>. [Accessed 30 Mayo 2015].
- [34] tk120404, "Stackoverflow," 2 Julio 2014. [Online]. Available: <http://stackoverflow.com/questions/21597797/format-error-while-installing-wireless-adapter>. [Accessed 15 Junio 2015].
- [35] E. Friedrich, "Introduction to Raspberry Pi," Artisan's Asylum, [Online]. Available: <http://raspberrypi-aa.github.io/session3/i2c-temp-pressure.html>. [Accessed 15 Junio 2015].
- [36] Davis Instruments, "Integrated Sensor Suite with," 22 Marzo 2006. [Online]. Available: [http://www.davisnet.com/product\\_documents/weather/manuals/VP2\\_Fan\\_Aspirated\\_ISS\\_Rev\\_C.pdf](http://www.davisnet.com/product_documents/weather/manuals/VP2_Fan_Aspirated_ISS_Rev_C.pdf). [Accessed 10 Mayo 2015].
- [37] TFV-Solar, "Ahorro energetico-Vida sostenible-Nuevas energias," [Online]. Available: [http://www.tiendafotovoltaica.es/epages/61359426.sf/es\\_ES/?ObjectPath=/Shops/61359426/Products/%22Davis%20Vantage%20Pro%202%20Aktiv%20Plus%206163EU%22](http://www.tiendafotovoltaica.es/epages/61359426.sf/es_ES/?ObjectPath=/Shops/61359426/Products/%22Davis%20Vantage%20Pro%202%20Aktiv%20Plus%206163EU%22). [Accessed 15 Junio 2015].
- [38] Zbiros, "Malinowo.net.pl," 7 Febrero 2015. [Online]. Available: <http://malinowo.net.pl/raspberry-pi-czujnik-opadow-deszczu-yl-83-skrypt-w-python-demon-mysql/>. [Accessed 30 Junio 2015].
- [39] S. p. w. station, «Softwarefun.org,» [En línea]. Available:

- <http://softwarefun.org/weatherstation>. [Último acceso: 14 07 2015].
- [40] «Softwarefun.org,» 2014. [En línea]. Available: <http://softwarefun.org/weatherstation>. [Último acceso: 20 Julio 2015].
- [41] Mosaic Documentation Web, «GPIO Electrical Specifications,» [En línea]. Available: <http://www.mosaic-industries.com/embedded-systems/microcontroller-projects/raspberry-pi/gpio-pin-electrical-specifications>. [Último acceso: 25 07 2015].
- [42] P. J. Vis, «Raspberry Pi GPIO: CMOS to TTL Logic Level Converter Circuit,» [En línea]. Available: [http://www.petervis.com/Raspberry\\_PI/CMOS\\_to\\_TTL\\_Logic\\_Level\\_Converter/CMOS\\_to\\_TTL\\_Logic\\_Level\\_Converter.html](http://www.petervis.com/Raspberry_PI/CMOS_to_TTL_Logic_Level_Converter/CMOS_to_TTL_Logic_Level_Converter.html). [Último acceso: 28 Julio 2015].

## ANEXOS

### Anexo A. Programa para la lectura de temperatura y humedad.

*nano dth11.c*

*Inicio del programa.*

```
//Librerias necesarias
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>

//Definicion de constantes
#define MAX_TIME 85
#define DHT11PIN 7
#define ATTEMPTS 5

//Definición un vector global
int dht11_val[5]={0,0,0,0,0};

////////////////////////////////////
//Función principal para leer los valores del sensor.
int dht11_read_val(){
    uint8_t lststate=HIGH;
    uint8_t counter=0;
    uint8_t j=0,i;
    for(i=0;i<5;i++){
        dht11_val[i]=0;
    }
    pinMode(DHT11PIN,OUTPUT);
    digitalWrite(DHT11PIN,LOW);
    delay(18);
    digitalWrite(DHT11PIN,HIGH);
    delayMicroseconds(40);
    pinMode(DHT11PIN,INPUT);
    for(i=0;i<MAX_TIME;i++){
        counter=0;
        while(digitalRead(DHT11PIN)==lststate){
            counter++;
            delayMicroseconds(1);
            if(counter==255){
                break;
            }
        }
        lststate=digitalRead(DHT11PIN);
        if(counter==255){
            break;
        }
    }
    //Las 3 primeras transiciones son ignoradas
    if((i>=4)&&(i%2==0)){
```

```

dht11_val[j/8]<=1;
if(counter>16){
dht11_val[j/8]=1;
}
j++;
}
}

// Se hace una suma de comprobación para ver si el dato es correcto. Si es así, se muestra
if((j>=40)&&(dht11_val[4]==((dht11_val[0]+dht11_val[1]+dht11_val[2]+dht11_val[3]& 0xFF))){
printf("%d,%d\n",dht11_val[0]*256+dht11_val[1],dht11_val[2]*256+dht11_val[3]);

return 1;
}else{
return 0;
}
}

////////////////////////////////////
//Empieza programa principal.
int main(void){
//Se establece el número de intentos que vamos a realizar
//la constante ATTEMPTS está definida arriba
int attempts=ATTEMPTS;

//Si la libreria wiringPi, ve el GPIO no está listo, se sale de la aplicación
if(wiringPiSetup()==-1){
exit(1);
}

while(attempts){
//Se intenta leer el valor del gpio, llamando a la función
int success = dht11_read_val();

//Si hay lectura con éxito, salimos del while, y se acaba el programa
if (success){
break;
}

//Si no lee con éxito, restamos 1, al número de intentos
attempts--;

//Espera de medio segundo antes del siguiente intento.
delay(500);
}
return 0;
}

```

## Anexo B. Código del anemómetro.

```

#By Lars-Martin Hejll
#http://softwarefun.org
#windData.py

```

```

#collects wind speed and direction
#Using Weather Sensor Assembly p/n 80422 Imported by Argent Data Systems
#from sparkfun.com
# one RPS = 1.492 MPH of wind rfactor (datasheet)

from time import sleep
import time
import RPi.GPIO as GPIO, time, os
from datetime import datetime
import math
import MySQLdb
#import connect

DEBUG = 0
counter = 0.0
finishtime = 0

rfactor = 1.492 #one RPS factor
samples = 20.0
speed = 0.0
directionPin = 18 #GPIO pin
speedPin = 23 #GPIO pin
state = False

#setup GPIO's
GPIO.setmode(GPIO.BCM)
GPIO.setup(speedPin, GPIO.IN)

#Speed messurment
def getSpeed():
    # Loop some seconds (samples) and record pulses
    counter = 0.0 #numbers of interrupt
    # finishtime is right now (clock time) + 100 real seconds, not
    # CPU cycles
    endTime = (int(time.time()) + samples)
    state = False
    while (int(time.time()) < endTime):
        if ( GPIO.input(speedPin) == True ):
            state = True #closed
            # wait for switch for open
            if ((state == True) and (GPIO.input(speedPin) == False)):
                # State is now open!
                state = False
                # count it!
                counter = counter + 1
    # counter is the total number of pulses during the sample time
    # speed in MPH

    speed = ((counter / samples) * rfactor)

    return (speed*0.44704)

def writeToDb(speed):
    # Inicia conexion al MySql

```

```

conn = MySQLdb.connect("localhost","root","tony","RASPBERRYPI")
cursor = conn.cursor()

params = [speed]
try:
    cursor.execute("INSERT INTO anemometro (date,speed) VALUE (NOW(),%s)",params)
    conn.commit()
except MySQLdb.Error, e:
    print "An error has occurred. %s" %e
finally:
    cursor.close()
    conn.close()
def main():
    speed = getSpeed()
    print ("Imprime Velocidad");
    print (speed);
    writeToDb(speed)
    print("Agregado a la base de datos OK");
try:
    main()
finally:
    print("Acaba la ejecucion");

#By Lars-Martin Hejll
#http://softwarefun.org
#windData.py

```

Se creó una tabla en la base de datos RASPBERRY para el almacenamiento de los datos suministrados por el sensor. El código para la realización de dicha tabla es el siguiente:

```

CREATE TABLE IF NOT EXISTS `anemometro` (
  `consec` int(11) NOT NULL AUTO_INCREMENT,
  `date` datetime NOT NULL,
  `speed` float NOT NULL,
  PRIMARY KEY (`consec`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Ano' AUTO_INCREMENT=8 ;

```

### Anexo C. Código implementado para el sensor YL-83.

```

#!/usr/bin/python
from __future__ import division
import RPi.GPIO as GPIO
from time import sleep
from datetime import datetime
import math
import MySQLdb

"""YL-83InsertRainMetherMysql.py: Insert measuring the amount of rainfall to MySQL database """
__author__ = "zbiros"
__copyright__ = "Copyright 2015, Malinowo.net.pl"

```

```

__credits__ = ["maros"]
__license__ = "GPL"
__version__ = "1.0.1"
__maintainer__ = "zbiros"
__email__ = "zbiros@malinowo.net.pl"
__status__ = "Development"

pin = 21
GPIO.setmode(GPIO.BCM) # set up BCM GPIO numbering
GPIO.setup(pin, GPIO.IN) # set GPIO21 = PIN 40 as input

count_tot_rain = 0
count_tot_norain = 0
counter_total = 0
write_cycle = 2 # 3600 seconds = 1 hour

# adds a record to the database
def insert_rain_reading (rain_value):

    conn = MySQLdb.connect("localhost","root","tony","RASPBERRYPI")
    cursor = conn.cursor()
    params = [rain_value]
    try:
        cursor.execute("INSERT INTO yl83 (date,rain) VALUE (NOW(),%s)",params)
        conn.commit()
    except MySQLdb.Error, e:
        print "An error has occurred. %s" %e
    finally:
        cursor.close()
        conn.close()

print "Anthony X1000 - Rain:" + `rain_value`
try:
    while True: # this will carry on until you hit CTRL+C
        if GPIO.input(pin):
            count_tot_norain = count_tot_norain + 1
            event = "no rain"
        else:
            count_tot_rain = count_tot_rain + 1
            event = "rain"
        count_tot = count_tot_norain + count_tot_rain
        print datetime.now().strftime("%Y-%m-%d [ %H ]") + "|" + event + "|" + `count_tot`

        if count_tot == write_cycle: # 1 hour
            if count_tot_norain == 0:
                perc_rain = 100
                perc_norain = 0
            elif count_tot_rain == 0:
                perc_rain = 0
                perc_norain = 100
            else:
                if count_tot_norain > count_tot_rain:
                    perc_norain = int(math.floor((100 * count_tot_norain) / write_cycle))
                    perc_rain = 100 - perc_norain

```

```

else:
    perc_rain = int(math.floor((100 * count_tot_rain) / write_cycle))
    perc_norain = 100 - perc_rain
    count_tot_norain = 0
    count_tot_rain = 0
    print "-----"
    print "Rain:" + `perc_rain` + "%|No Rain:" + `perc_norain` + "%"
    print "-----"

    #add to mysql
    insert_rain_reading(perc_rain)

    #break;
    sleep(1)      # wait 1 seconds
finally:         # this block will run no matter how the try block exits
    GPIO.cleanup()    # clean up after yourself

```

## Anexo D. Aplicación Web.

Con el fin de que el usuario final de la estación meteorológica móvil pueda visualizar de forma rápida y sencilla la información suministrada por los sensores, se realizó una página en la cual se mostrarán los datos reales medidos en el momento. Dicha página se elaboró con JQuery para los estilos y Highcharts para las gráficas, con el siguiente código en PHP:

```

<?php
ini_set('display_errors', true);
error_reporting(E_ALL & ~E_NOTICE);
class GraphPilotoSensor
{

    private static $cConnec = NULL;
    private static $cServer = "localhost";
    private static $cUserxx = "root";
    private static $cPasswd = "tony";
    private static $cDataBx = "RASPBERRYPI";

    # Funcion constructora de la clase -----
    function __construct()
    {
        try
        {
            if( $_REQUEST["opcion"])
            {
                switch ($_REQUEST["opcion"]) {
                    case '1': echo self::GraphsDth11(); break;
                    case '2': echo self::GraphsBMP180(); break;
                    case '3': echo self::GraphsYI83(); break;
                    case '4': echo self::GraphsAnemometro(); break;
                }
            }
            exit();
        }
    }
}

```



```

else
{
    self::BodyOpen();
    self::Title();
    self::TableViewLastData();
    self::Filters();
    self::Tabs();
    self::BodyClose();
}
}
catch(Exception $e)
{
    echo "<pre>"; print_r($e); echo "</pre>";
}
}

private function TableViewLastData()
{
    $Dth11 = self::GraphsDth11Query( true );
    $YI83 = self::GraphsYI83Query ( true );
    $BMP180 = self::GraphsBMP180Query ( true );
    $Anemometro = self::GraphsAnemometroQuery ( true );

    $mHtml = '<table class="ui-tabs ui-widget ui-widget-content ui-corner-all"><tr><td><table
class="ui-tabs-nav ui-helper-clearfix ui-widget-header ui-corner-all">';
    $mHtml .= '<tr>';
    $mHtml .= '<td colspan="4" align="center">Datos Actuales</td>';
    $mHtml .= '</tr>';
    $mHtml .= '<tr>';
    $mHtml .= '<td class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-
all">Sensor</td>';
    $mHtml .= '<td class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-
all">Fecha Registro</td>';
    $mHtml .= '<td class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-
all">Valor1</td>';
    $mHtml .= '<td class="ui-tabs-nav ui-helper-reset ui-helper-clearfix ui-widget-header ui-corner-
all">Valor2</td>';
    $mHtml .= '</tr>';
    $mHtml .= '<tr>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">Temperatura -
Humedad</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $Dth11[0]["fec_regist"]. '</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $Dth11[0]["val_humeda"]. '</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $Dth11[0]["val_temper"]. '</td>';
    $mHtml .= '</tr>';
    $mHtml .= '<tr>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">Presión
Atmosférica</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $BMP180[0]["fecha"]. '</td>';

```

```

    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $BMP180[0]["val_pres"].'</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">&nbsp;</td>';
    $mHtml .= '</tr>';
    $mHtml .= '<tr>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">Lluvia</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">'. $Y183[0]["date"].'</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">'. $Y183[0]["rain"].'</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">&nbsp;</td>';
    $mHtml .= '</tr>';

    $mHtml .= '<tr>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">Velocidad Viento</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $Anemometro[0]["date"].'</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-
all">'. $Anemometro[0]["speed"].'</td>';
    $mHtml .= '<td class="ui-tabs ui-widget ui-widget-content ui-corner-all">&nbsp;</td>';
    $mHtml .= '</tr>';
    $mHtml .= '</table></td></tr></table><br>';

    echo $mHtml;
}
private function BodyOpen()
{
    $mHtml = '<html>
    <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Estación Metereológica</title>
    <script type="text/javascript" src="js/Jquery1.js"></script>
    <script type="text/javascript" src="js/Jquery2.js"></script>
    <script src="highcharts.js"></script>
    <script src="exporting.js"></script>

    <link rel="stylesheet" type="text/css" href="css/Jcss1.css">
    <style type="text/css">
    body {
        background-image: url("BackGreen1.jpg");
        background-size: 100% ;
        background-repeat: no-repeat;
    }
    </style>';
    $mHtml .= self::Javascript();
    $mHtml .= '</head>
    <body>';
    echo $mHtml;
}

private function Title()
{
    $mHtml = '<table class="ui-tabs ui-widget ui-widget-content ui-corner-all" align="center">

```

```

        <tr><td><table class="ui-tabs-nav ui-helper-clearfix ui-widget-header ui-corner-all">;
        $mHtml .= '<tr>';
        $mHtml .= ' <td colspan="4" align="center" style="font-size: 30px;">ESTACIÓN
METEREOLÓGICA - UNIVERSIDAD PILOTO DE COLOMBIA </td>';
        $mHtml .= '</tr>';
        $mHtml .= '</table></td></tr></table><br>';

    echo $mHtml;
}

private function Javascript()
{
    $mJavascript = "<script type='text/javascript'>
        $(function() {
            $('#tabs').tabs();
            $('#fec_ini,#fec_fin').datepicker({
                dateFormat: 'yy-mm-dd'
            });

            $('#tabs').tabs({
                beforeLoad: function( event, ui ) {
                    ui.jqXHR.fail(function() {
                        ui.panel.html(
                            'No se puede cargar el complemento' );
                    });
                }
            });
        });
    </script>";
    return $mJavascript;
}

private function Filters()
{
    $mHtml = '<form name="filtros" id="filtrosID" method="post" action="?" >
        <table width="auto" align="center" cellpadding="4" border="1" class="ui-tabs ui-widget ui-
widget-content ui-corner-all" >;
        $mHtml .= '<tr>';
        $mHtml .= '<td>';
        $mHtml .= '<div>';
        $mHtml .= '<table width="100%" class="ui-tabs-nav ui-helper-clearfix ui-widget-header ui-
corner-all" >;
        $mHtml .= '<tr>';
        $mHtml .= '<td align="center">Filtro Por Fecha</td>';
        $mHtml .= '</tr>';
        $mHtml .= '</table>';
        $mHtml .= '<table width="100%" class="ui-tabs-nav ui-helper-clearfix ui-widget-header ui-
corner-all" >;
        $mHtml .= '<tr>';
        $mHtml .= '<td>Fecha Inicial:</td>';
    </form>';
}

```

```

        $mHtml .= '<td><input type="text" name="fec_ini" id="fec_ini"
value="'.$_REQUEST["fec_ini"].'" /></td>';
        $mHtml .= '</tr>';
        $mHtml .= '<tr>';
        $mHtml .= '<td>Fecha Final:</td>';
        $mHtml .= '<td><input type="text" name="fec_fin" id="fec_fin"
value="'.$_REQUEST["fec_fin"].'" /></td>';
        $mHtml .= '</tr>';
        /*$mHtml .= '<tr>';
        $mHtml .= '<td>Datos a mostrar:</td>';
        $mHtml .= '<td><input type="text" name="fec_fin" id="fec_fin"
value="'.$_REQUEST["fec_fin"].'" /></td>';
        $mHtml .= '</tr>';
        */
        $mHtml .= '<tr>';
        $mHtml .= '<td align="center" colspan="2"><input type="submit" value="Aceptar"
name="Aceptar" id="AceptarID" /></td>';
        $mHtml .= '</tr>';
        $mHtml .= '</table>';
        $mHtml .= '</div>';
        $mHtml .= '</td>';
        $mHtml .= '</tr>';
        $mHtml .= '</table></form><br>';
        echo $mHtml;
    }

    private function BodyClose()
    {
        $mHtml = '</body></html>';
        echo $mHtml;
    }

    private function Tabs()
    {
        {
            try
            {
                $mHtml = '<div id="tabs">';
                $mHtml .= ' <ul>
                    <li><a href="index.php?opcion=1&fec_ini='.$_REQUEST["fec_ini"].'&fec_fin='.$_REQUEST["fec_fin"].'">Temperatura - Humedad</a></li>
                    <li><a href="index.php?opcion=2&fec_ini='.$_REQUEST["fec_ini"].'&fec_fin='.$_REQUEST["fec_fin"].'">Presión Atmosférica</a></li>
                    <li><a href="index.php?opcion=3&fec_ini='.$_REQUEST["fec_ini"].'&fec_fin='.$_REQUEST["fec_fin"].'">Lluvia</a></li>
                    <li><a href="index.php?opcion=4&fec_ini='.$_REQUEST["fec_ini"].'&fec_fin='.$_REQUEST["fec_fin"].'">Velocidad Viento</a></li>
                </ul>';
                $mHtml .= '<div id="tabs-1">';
                $mHtml .= '</div>';

                $mHtml .= '</div>';
                echo $mHtml;
            }
        }
    }

```

```

catch(Exception $e)
{
    echo "<pre> Error tabs:<br>"; print_r($e); echo "</pre>";
}
}

private function GraphsDth11Query( $mGeneral = true)
{
    $mQuery = 'SELECT *
        FROM '.self::$cDataBx.'.dth11
        WHERE 1 = 1 AND
        fec_regist >= '.( $_REQUEST["fec_ini"] ? "".$_REQUEST["fec_ini"]." 00:00:00"." : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').' AND
        fec_regist <= '.( $_REQUEST["fec_fin"] ? "".$_REQUEST["fec_fin"]." 23:59:59"." : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').'
        ORDER BY fec_regist DESC '.( $mGeneral == false ? 'LIMIT 20' : 'LIMIT 1') ;
    return $mData = self::Conection($mQuery);
}

# Funcion para graficar Temperatura - Humedad
private function GraphsDth11()
{
    try
    {

        $mData = self::GraphsDth11Query( false );

        $mDatax = array();
        foreach ($mData AS $mKey => $mValue):
            $mDatax["fecha"][] = $mValue["fec_regist"];
            $mDatax["humed"][] = $mValue["val_humeda"];
            $mDatax["tempe"][] = $mValue["val_temper"];
        endforeach;

        #echo "<pre>Data dth11:<br>"; print_r($mDatax); echo "</pre>";

        $mHtml .= '<div id="container" style="border:5px solid black; box-shadow: -4px 6px 29px 14px
        rgba(0,0,0,0.7); border-radius: 10px; min-width: 310px; height: 400px; margin: 0 auto" data-
        highcharts-chart="0">
            </div>';

        $mHtml .= "<script>
            $(function () {
                $('#container').highcharts({
                    title: {
                        text: 'Estación Meteorológica',
                        x: -20 //center
                    },
                    subtitle: {
                        text: 'Universidad Piloto de Colombia<br>http://www.highcharts.com/about',
                        x: -20
                    },
                    xAxis: {
                        categories: [". "" . @join($mDatax["fecha"], "", ""). "" . " ]

```

```

    },
    yAxis: {
        title: {
            text: 'Temperatura - Humedad'
        },
        plotLines: [{
            value: 0,
            width: 1,
            color: '#808080'
        }]
    },
    tooltip: {
        valueSuffix: "
    },
    legend: {
        layout: 'vertical',
        align: 'right',
        verticalAlign: 'middle',
        borderWidth: 0
    },
    series: [{
        name: 'Humedad (%)',
        data: [ ". @join($mDatax['humed'], ',')." ]
    }, {
        name: 'Temperatura (°C)',
        data: [ ". @join($mDatax['tempe'], ',')." ]
    }]
});

```

```

});
</script> ";

```

```

    return $mHtml;
}
catch(Exception $e)
{
    echo "<pre>Error GraphsDth11: "; print_r($e); echo "</pre>";
}
}

```

```

private function GraphsYl83Query( $mGeneral = true)
{
    $mQuery = 'SELECT *
        FROM '.self::$cDataBx.'.yl83
        WHERE 1 = 1 AND
            date >= '.( $ _REQUEST["fec_ini"] ? "".$ _REQUEST["fec_ini"]." 00:00:00"."" : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').' AND
            date <= '.( $ _REQUEST["fec_fin"] ? "".$ _REQUEST["fec_fin"]." 23:59:59"."" : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').'
        ORDER BY date DESC '.( $mGeneral == false ? 'LIMIT 20' : 'LIMIT 1');

    return $mData = self::Conection($mQuery);
}

```

```

}

# Funcion para graficar Lluvia
private function GraphsYl83()
{
    try
    {

        $mData = self::GraphsYl83Query( false );

        $mDatax = array();
        foreach ($mData AS $mKey => $mValue):
            $mDatax["fecha"][] = $mValue["date"];
            $mDatax["rainx"][] = $mValue["rain"];
        endforeach;

        #echo "<pre>Data dth11:<br>"; print_r($mDatax); echo "</pre>";

        $mHtml .= '<div id="containerYl83" style="border:5px solid black; box-shadow: -4px 6px 29px
14px rgba(0,0,0,0.7); border-radius: 10px; min-width: 310px;max-width: 100%; height: 400px;
margin: 0 auto" data-highcharts-chart="0">
            </div>';

        $mHtml .= "<script>
            $(function () {
                $('#containerYl83').highcharts({
                    title: {
                        text: 'Estación Meteorológica',
                        x: -20 //center
                    },
                    subtitle: {
                        text: 'Universidad Piloto de Colombia<br>http://www.highcharts.com/about',
                        x: -20
                    },
                    xAxis: {
                        categories: [". "" . @join($mDatax["fecha"], ", "). "" . " ]
                    },
                    yAxis: {
                        title: {
                            text: 'Lluvia'
                        },
                        plotLines: [{
                            value: 0,
                            width: 1,
                            color: '#808080'
                        }]
                    },
                    tooltip: {
                        valueSuffix: "
                    },
                    legend: {
                        layout: 'vertical',
                        align: 'right',
                        verticalAlign: 'middle',

```

```

        borderWidth: 0
    },
    series: [{
        name: 'Lluvia (%)',
        data: [ ". @join($mData['rainx'], ',')." ]
    }]
});

// redimensiona div de GraphsYl83
$('#containerYl83').css({ width: '100%' }).resizable();

});
</script> ";

return $mHtml;
}
catch(Exception $e)
{
    echo "<pre>Error GraphsYl83: "; print_r($e); echo "</pre>";
}
}

private function GraphsBMP180Query( $mGeneral = true)
{
    $mQuery = 'SELECT *
        FROM '.self::$cDataBx.'.bmp180
        WHERE 1 = 1 AND
        fecha >= '.( $_REQUEST["fec_ini"] ? "".$_REQUEST["fec_ini"]." 00:00:00"." : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').' AND
        fecha <= '.( $_REQUEST["fec_fin"] ? "".$_REQUEST["fec_fin"]." 23:59:59"." : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').'
        ORDER BY fecha DESC '.( $mGeneral == false ? 'LIMIT 20' : 'LIMIT 1');

    return $mData = self::Conection($mQuery);
}

# Funcion para graficar presion atmosferica
private function GraphsBMP180()
{
    try
    {

        $mData = self::GraphsBMP180Query( false );

        $mData = array();
        foreach ( $mData AS $mKey => $mValue):
            $mData["fecha"][] = $mValue["fecha"];
            $mData["val_pres"][] = $mValue["val_pres"];
        endforeach;

        #echo "<pre>Data dth11:<br>"; print_r($mData); echo "</pre>";
    }
}

```



```

    $mHtml .= '<div id="containerBMP180" style="border:5px solid black; box-shadow: -4px 6px
29px 14px rgba(0,0,0,0.7); border-radius: 10px; min-width: 310px;max-width: 100%; height: 400px;
margin: 0 auto" data-highcharts-chart="0">
        </div>';

```

```

$mHtml .= "<script>
    $(function () {
        $('#containerBMP180').highcharts({
            title: {
                text: 'Estación Meteorológica',
                x: -20 //center
            },
            subtitle: {
                text: 'Universidad Piloto de Colombia<br>http://www.highcharts.com/about',
                x: -20
            },
            xAxis: {
                categories: [". "" . @join($mDatax["fecha"], "", ""). "" . " ]
            },
            yAxis: {
                title: {
                    text: 'Presión'
                },
                plotLines: [{
                    value: 0,
                    width: 1,
                    color: '#808080'
                }]
            },
            tooltip: {
                valueSuffix: "
            },
            legend: {
                layout: 'vertical',
                align: 'right',
                verticalAlign: 'middle',
                borderWidth: 0
            },
            series: [{
                name: 'Presión (%)',
                data: [ ". @join($mDatax['val_pres'], ','). " ]
            }]
        });

```

```

// redimensiona div de GraphsYl83

```

```

$('#containerYl83').css({ width: '100%' }).resizable();

```

```

});
</script> ";

```

```

    return $mHtml;
}
catch(Exception $e)

```

```

{
    echo "<pre>Error GraphsBMP180: "; print_r($e); echo "</pre>";
}
}

private function GraphsAnemometroQuery( $mGeneral = true)
{
    $mQuery = 'SELECT *
        FROM '.self::$cDataBx.'.anemometro
        WHERE 1 = 1 AND
        date >= '.( $_REQUEST["fec_ini"] ? "$_REQUEST["fec_ini"]." 00:00:00"." : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').' AND
        date <= '.( $_REQUEST["fec_fin"] ? "$_REQUEST["fec_fin"]." 23:59:59"." : '
DATE_SUB( NOW(), INTERVAL 1 DAY ) ').'
        ORDER BY date DESC '.( $mGeneral == false ? 'LIMIT 20' : 'LIMIT 1');

    return $mData = self::Conection($mQuery);
}

# Funcion para graficar presion atmosferica
private function GraphsAnemometro()
{
    try
    {

        $mData = self::GraphsAnemometroQuery( false );

        $mDatax = array();
        foreach ( $mData AS $mKey => $mValue):
            $mDatax["fecha"][] = $mValue["date"];
            $mDatax["speed"][] = $mValue["speed"];
        endforeach;

        #echo "<pre>Data dth11:<br>"; print_r($mDatax); echo "</pre>";

        $mHtml .= '<div id="containerAnemometro" style="border:5px solid black; box-shadow: -4px
6px 29px 14px rgba(0,0,0,0.7); border-radius: 10px; min-width: 310px;max-width: 100%; height:
400px; margin: 0 auto" data-highcharts-chart="0">
        </div>';

        $mHtml .= "<script>
        $(function () {
            $('#containerAnemometro').highcharts({
                title: {
                    text: 'Estación Meteorológica',
                    x: -20 //center
                },
                subtitle: {
                    text: 'Universidad Piloto de Colombia<br>http://www.highcharts.com/about',

```

```

        x: -20
    },
    xAxis: {
        categories: ["". @join($mData["fecha"], ", "). "" ]
    },
    yAxis: {
        title: {
            text: 'Velocidad Viento'
        },
        plotLines: [{
            value: 0,
            width: 1,
            color: '#808080'
        }]
    },
    tooltip: {
        valueSuffix: "
    },
    legend: {
        layout: 'Velocidad Viento',
        align: 'right',
        verticalAlign: 'middle',
        borderWidth: 0
    },
    series: [{
        name: 'Velocidad Viento (Km/h)',
        data: [ ". @join($mData['speed'], ','). " ]
    },]
});

// redimensiona div de GraphsYl83
$("#containerYl83").css({ width: '100%' }).resizable();

});
</script> ";

return $mHtml;
}
catch(Exception $e)
{
    echo "<pre>Error GraphsAnemometro: "; print_r($e); echo "</pre>";
}
}

# -----
# -----
# -----
# -----

private function Conection($mQuery)

```

```

{
  try
  {
    # Inicia la conexion a BD -----
    $mLink = mysql_connect( self::$cServer, self::$cUserxx, self::$cPasswd );
    if(!$mLink) {
      throw new Exception("Error Conexion a BD: ".mysql_error($mLink), 9001);
    }
    # Selecciona la BD a trabajar
    $mBD = mysql_select_db(self::$cDataBx, $mLink);
    if(!$mBD){
      throw new Exception("Error Select BD: ".mysql_error($mBD), 9001);
    }

    # Hace la query -----
    $mExecute = mysql_query($mQuery, $mLink);
    $mArray= array();
    while ( $mRow = mysql_fetch_array($mExecute, MYSQL_ASSOC)) {
      $mArray[] = $mRow;
    }
    # cierra la conexion de BD
    mysql_close($mLink);
    return $mArray;
  }
  catch(Exception $e)
  {
    return array("cod_error" => "9001", "msg_error" => $e -> getMessage());
  }
  return array("cod_error" => "1000", "msg_error" => "Todo OK BD :)");
}

}
$mGraph = new GraphPilotoSensor();

?>

```

## Anexo E. Base de datos

Para el almacenamiento de la información se trabajó con el motor de base de datos Mysql nuevamente, ya que este es compatible con los lenguajes de programación utilizados para la lectura de los datos arrojados por los sensores.

Se instalaron las siguientes librerías para el trabajo con el motor de base de datos mediante la ejecución de los siguientes comandos:

```

sudo apt-get update
sudo apt-get upgrade
sudo apt-get install libmysqlclient-dev

```

Una vez incluidas las librerías, se modificó el programa para el almacenamiento de la información en la base de datos ejecutando el siguiente comando:

```
#include <mysql/mysql.h>
```

Adicional se realizó las creaciones de un usuario root y una contraseña, solicitadas por el motor una vez instalado; al momento de la creación de las credenciales se creó una base de datos llamada RASPBERRYPI y las respectivas tablas según el número variables físicas a medir; todo esto se realizó mediante las siguientes líneas de código:

```
$ mysql -u root -p
Enter password:
mysql> CREATE DATABASE temps
mysql> USE temps
mysql> CREATE USER 'monitor'@'localhost' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON temps.* TO 'monitor'@'localhost'
mysql> FLUSH PRIVILEGES;
mysql> quit
```

## Creación de Tablas

Es necesaria la creación de una tabla por cada sensor, la línea de comandos utilizada fue la siguiente:

### BMP180

```
-- phpMyAdmin SQL Dump
-- version 3.4.11.1deb2+deb7u1
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
-- Base de datos: `RASPBERRYPI`
-- Estructura de tabla para la tabla `bmp180`
CREATE TABLE IF NOT EXISTS `bmp180` (
  `num_consec` int(11) NOT NULL AUTO_INCREMENT,
  `fecha` datetime NOT NULL,
  `val_temper` float NOT NULL,
  `val_pres` float NOT NULL,
  PRIMARY KEY (`num_consec`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

-- Volcado de datos para la tabla `bmp180`

INSERT INTO `bmp180` (`num_consec`, `fecha`, `val_temper`, `val_pres`) VALUES
(6, '2015-06-10 00:00:00', 29.1, 75183),
(7, '2015-06-10 00:00:00', 28.8, 75187);
```

## DHT11

```
-- phpMyAdmin SQL Dump
-- version 3.4.11.1deb2+deb7u1
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
-- Base de datos: `RASPBERRYPI`
-- Estructura de tabla para la tabla `dth11`

CREATE TABLE IF NOT EXISTS `dth11` (
  `num_consec` int(11) NOT NULL AUTO_INCREMENT,
  `fec_regist` datetime NOT NULL,
  `val_humeda` float NOT NULL,
  `val_temper` float NOT NULL,
  PRIMARY KEY (`num_consec`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Registros Humedad y Calor'
AUTO_INCREMENT=937;
```

## Anemómetro

```
CREATE TABLE IF NOT EXISTS `anemometro` (
  `consec` int(11) NOT NULL AUTO_INCREMENT,
  `date` datetime NOT NULL,
  `speed` float NOT NULL,
  PRIMARY KEY (`consec`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Ano' AUTO_INCREMENT=8;
```

## Detector de lluvia

```
sudo mysql -u root -p
USE RASPBERRYPI;
CREATE TABLE y183 (
  date DATETIME NOT NULL,
  rain INT NULL);
```